ENERGY MANAGEMENT ENGINEERING

A Predictive Energy Management System incorporating an Adaptive Neural Network

for

the Direct Heating of Domestic and Industrial Fluid Mediums

A thesis presented for the degree of

Doctor of Philosophy

in Electrical and Electronic Engineering

at the University of Canterbury,

Christchurch, New Zealand.

February 2000

Herman Wezenberg

B.E.(Hons), M.I.E.E., C.Eng.

Abstract

The objective of this research project is to improve the control and provide a more cost-efficient operation in the direct heating of stored domestic or industrial fluid mediums; such to be achieved by means of an intelligent automated energy management system.

For the residential customer this system concept applies to the hot water supply as stored in the familiar hot water cylinder; for the industrial or commercial customer the scope is considerably greater with larger quantities and varieties of fluid mediums. Both areas can obtain significant financial savings with improved energy management. Both consumers and power supply and distribution companies will benefit with increased utilisation of cheaper 'off-peak' electricity; reducing costs and spreading the system load demand. The project has focussed on domestic energy management with a definite view to the wider field of industrial applications.

Domestic energy control methodology and equipment has not significantly altered for decades. However, computer hardware and software has since then flourished to an unprecedented proportion and has become relatively cheap and versatile; these factors pave the way for the application of computer technology in this area of great potential. The technology allows the implementation of a 'hot water energy management system', which makes a forecast of the hot water demand for the next 24 hours and proceeds to provide this demand in the most efficient manner possible. In the (near) future, the system, known as FEMS for *Fluid Energy Management System*, is able to take advantage and in fact will promote the use of a retail 'dynamic spot price tariff'.

FEMS is a combination of hardware and software developed to replace the existing cylinder thermostat, take care of the necessary data-acquisition and control the cylinder's total energy instead of it's (single point) temperature. This provides, besides heating cost reduction, a greater accuracy, a degree of flexibility, improved feedback, legionella inhibition, and a diagnostic capability. To the domestic consumer the latter three items are of greatest relevance.

The crux of the system lies in its predictive ability. Having explored the more conventional alternatives, a suitable solution was found in the utilisation of the Elman recurrent neural networks, which focus on the temporal characteristics of the hot water demand time series and are able to adapt to changing environments, coping with the presence of any non-linearity and noise in the data.

Prior to developing FEMS a study was made of the basic fluid behaviour in medium and high pressure *domestic* hot water cylinders, an area not well-covered to date and of interest to engineers and manufacturers alike. For this step data acquisition equipment and software was purposely created. The control software plus equipment were combined into a fully automated test system with minimal operator input, allowing a large amount of data to be gathered over a period measured in months. A similar system was subsequently used to collect actual hot water demand data from a residential family, and in fact forms the basis for FEMS.

Finally an enhanced version of FEMS is discussed and it is shown how the system is able to output multiple prediction and utilise varying tariff rates.

Ter Nagedachtenis Aan Mijn Vader,

Hermann Johan Wezenberg

(1928 – 1999)

Researchers can be likened to the classical blind men feeling different parts of an elephant. They have a lot of information but are still not certain what the elephant looks like. However, they do know that it's big.

Acknowledgements

This is the bit where I express my gratitude and acknowledge the support, so here it is:

My long suffering wife Julie, what can I say ..... couldn't have done it without you.

My supervisor Associate Professor Michael Dewe (love the title) for his patience and understanding in the delivery of this thesis.

Michelle and Stephan for putting up with a grumpy papa (I'll try not to hog the PC in future).

The technicians: specifically Ken, Dermot and Peter for the cylindrical bits and Pieter for the bytes.

And finally the guys who used to make up the Domestic Energy Management Research Group, now long gone; Stephen Hunt, Sikander Pathan (whither did he go?) and Peter Johnston. Het ga je goed, jongens.

CONTENTS

What does the reader do when he wishes to see in what the precise likeness or difference of two objects lies? He transfers his attention as rapidly as possible, backwards and forwards, from one to the other. The rapid alteration of consciousness shakes out, as it were, the points of difference or agreement, which would have slumbered forever unnoticed if the consciousness of the objects compared had occurred at widely distant periods of time.

What does the scientific man do when he searches for the reason or law embedded in a phenomenon? He deliberately accumulates all the instances he can find which have any analogy to that phenomenon; and by simultaneously filling his mind with them all, he frequently succeeds in detaching from the collection the peculiarity which he was unable to formulate in the one alone; even though that one had been preceded in his former experience by all those with which he now at once confronts it.

William James, 1890,

(1842-1910), American philosopher and psychologist, who developed the philosophy of pragmatism.

Contributions

The main contributions of this thesis are:

- Introducing the concept and feasibility of a fluid oriented energy management system (FEMS). This is a black box approach to increased energy efficiency in domestic and industrial situations.
- A detailed analysis of the behaviour of hot water in a domestic water cylinder during heating, draw-off and standing, and its application to Fluid Energy Management.
- A detailed description of the software written and hardware required for the Control and Data Acquisition system.
- A comparison of approaches to discrete time series predictions using statistical methods. Prediction is a key function of the FEMS.
- An analysis of forecasting with the Maximum Entropy method (MEM), with emphasis on a linear prediction system suitable for FEMS.
- Description of the software written for MEM forecasting.
- The introduction and discussion of Artificial Neural Networks (ANN) and a variety of typical applications including linear/non-linear forecasting, and indicating some of the challenges faced in actual practise.
- A systematic presentation of the current neural network models utilised for discrete time series prediction.
- A presentation of some the latest developments in the field of energy management systems, with particular focus on the domestic market.
- A description of the software written and the hardware required for FEMS, with justification for selecting the Recurrent Neural Network (RNN) prediction format.
- The acquisition and presentation of domestic consumer hot water demand and time-of-use data, accompanied by the multiple time series prediction results and explanation thereof.
- The introduction and discussion of the energy market, spot price tariffs and the latest developments in New Zealand with emphasis on the domestic consumer.
- The intended use of an enhanced version of FEMS for multiple predictions in a 24 hour period (using multiple time series), allowing exploitation of static or dynamic variable tariffs for improved energy management and demand-side load spreading.

Glossary

A	Amps
ANN	Artificial Neural Network
BEMS	Building Energy Management System
BPTT	Back-Propagation Through Time
C	Capacitor
'C'	A programming language
DI	Digital Input
DO	Digital Output
DSM	Demand Side Management
ECNZ	Electricity Corporation of New Zealand
EMS	Energy Management System
ESA	Electricity Supply Authorities
Farad	A unit of capacitance
FEMS	Fluid Energy Management System
FGS	Frasconi-Gori-Soda
GDP	Gross Domestic Product
GWh	GigaWatt hour
GUI	General User Interface
HME	Hierarchical Mixture of Experts
HW	Hot Water
IC	Integrated Circuit
I	Electrical current
IN	Interconnection Network
I/O	Input/Output
kJ	kiloJoules
kW	kiloWatt
kWh	kiloWatt hour
LAN	Local Area Network
LED	Light Emitting Diode
LP	Linear Prediction
LSI	Large Scale Integration
mA	milli-Amp
mV	milli-Volt
MARIA	Metering and Reconciliation Industry Agreement
MEM	Maximum Entropy Model
MLP	Multi-Layer Perceptron
MRE	Mean Relative Error
MS-TDNN	Multi State – Time Delay Neural Network
MW	MegaWatt
NZEM	New Zealand Electricity Marketplace
Ohms	A unit of resistance
PC	Personal Computer
PCB	Printed Circuit Board
PCL-814	A specialised I/O board manufactured by LabDas
PJ	Peta-Joules (10^{15} Joules)

R	Resistor
RBF	Radial Basis Function
RHONN	Recurrent High-Order Neural Network
RMS	Root-Mean-Square (error)
RNN	Recurrent Neural Network
RTRL	Real-Time Recurrent Learning
SWD	Sequential Waveform Distortion
TD	Temporal Difference
TDD	Truncated Temporal Difference
TDNN	Time Delay Neural Network
Th	Thermistor
T&D	Transmission and Distribution
V	Volt
VLSI	Very Large Scale Integration
WEM	Wholesale Electricity Marketplace
WEMS	Wholesale Electricity Marketplace Study
W&Z	William and Zipser
^{o}C	Degrees Celsius
^{o}K	Degrees Kelvin

Citations

A citation search of the Conference paper produced by H.W. and M.B.D. for the ICNN'95, entitled "*Adaptive neural networks for tariff forecasting and energy management*" (as can be found in: Proc. of the IEEE Int. Conf. ICNN'95, vol.2, Perth, 1995, p877-81) produced two references;

- Kaloghirou, S.A., Neocleous, C.C., Shizas, C.N., (1998), Artificial neural networks for modeling the start-up of a solar steam generator, *Applied Energy*, vol.60, issue 2, 89 – 100.
- Kaloghirou, S.A., (2000), Applications of artificial neural networks for energy systems, *Applied Energy*, vol.67, issue 1.

Chapter 1. Introduction

1.1 Background

Electricity, unlike gas and water, cannot be economically stored in large quantities. It must therefore be generated the instant the customer demands its use. Power demand typically varies widely throughout the day. In order to provide a reliable service, electricity suppliers must carry *excess* generating and distribution capacity to cope with the periods of high demand. To smooth demand, the suppliers charge a lower tariff during the off-peak periods. At present the *domestic* off-peak tariff in a country such as New Zealand generally occurs during the hours of 11 p.m. to 7 a.m.

A large proportion of the total value of a typical household electricity bill in New Zealand is due to the demand of the domestic hot water cylinder where typically 180 litres of hot water at a temperature of around 70°C is stored on a continuous basis. Recent figures put out by the EECA (Energy Efficiency and Conservation Authority) shows that a typical family spends about 45% of its energy bill on hot water (*Figure 1.1*).

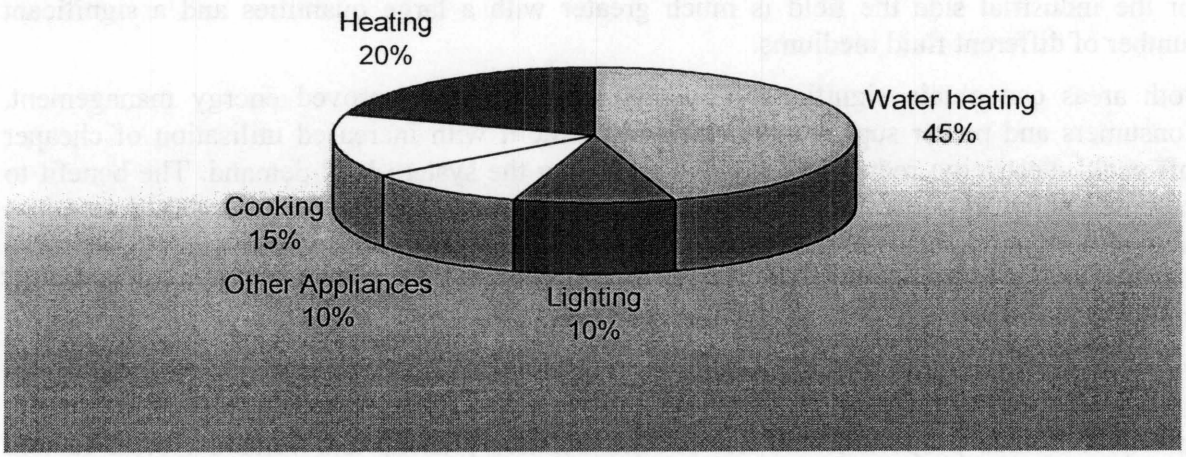


***Figure 1.1* – Usage of energy in the home.**

When put against a background of *rising electricity prices* and the possibility of a *dynamic spot-price tariff* system there would appear to be considerable merit in managing this form of energy storage system in a more efficient manner.

The energy used for *domestic hot-water heating* not only increases in cost as unit fuel cost rise but also in importance as local power distributors move towards a tariff-based market structure. In such a market the consumer will pay a varying price for the electricity used depending on the total network demand at the time of consumption. As hot-water heating together with space heating form the main bulk of the domestic energy bill it would seem advantageous to develop a system which will make efficient economic use of the lower electricity tariffs while at the same time, from the viewpoint of the supply authority, minimizing the maximum demand levels.

Efficient usage in the mind of the domestic consumer means a reduction in appliance power consumption and therefore a lesser amount that has to be paid for the electricity used. Local

power distribution companies are interested in lessening domestic power consumption as they attempt to reduce peaks in the daily electricity demand. Apart from actually improving the efficiency of the electrical appliances concerned the only remaining method of reducing energy consumption at peak periods, while simultaneously satisfying both consumer and power company, is to encourage utilisation of power demanding apparatus in the periods of low network load by direct coupling to low tariff prices.

New Zealand is undergoing some far-reaching changes in its electricity pricing structure. *Local* distribution companies such as Southpower (in the Canterbury province) are looking at introducing a variable tariff system that will reflect in its spot-price the demand of electricity on a half-hour or hourly basis. The lower tariff prices will therefore be available in different, and probably varying, time slots during a 24 hour period. Consumers will find it increasingly difficult to keep track of economically suitable spot-price time allotments and some form of intelligent automation in switching a major appliance such as the hot water cylinder on and off would be both beneficial and profitable.

1.2 Purpose of the study

The objective of this study is to significantly contribute to, and advance, a new control technique for domestic/industrial energy fluid storage, and thus provide a more efficient control over the direct heating by electrical means of stored domestic or industrial fluid mediums. For the domestic side this applies to the hot water supply as stored in a cylinder; for the industrial side the field is much greater with a large quantities and a significant number of different fluid mediums.

Both areas can obtain significant financial savings with improved energy management. Consumers and power supply authorities will benefit with increased utilisation of cheaper 'off-peak' electricity; reducing costs and spreading the system load demand. The benefit to the environment is self-evident; a country's peak demand for electricity largely determines the number and capacity of generating stations built. The trend over the years has been a steady increase in demand; being able to reduce peak load means that the construction of another powerstation can be delayed or maybe even avoided.

It is noted that domestic energy control methodology and equipment has not significantly altered for decades. However, computer hardware and software has since then flourished to an unprecedented proportion and has become relatively cheap and versatile; these factors pave the way for the application of computer technology in this area of great potential. The technology allows the implementation of a 'fluid energy management system' which takes advantage of a 'dynamic spot price tariff'. The study will focus on domestic energy management where the fluid is (hot) water, but the implications with regards to the wider field of industrial applications should not be overlooked.

1.3 Statement of the problem

Is it possible to forecast hot water demand, or the demand of any similar heated medium, with a minimum of *a priori* data, and utilise this information in reducing the energy costs for the consumer (domestic or industrial) by taking advantage of varying electricity tariff rates and simultaneously curtail or spread more evenly the peak load demand, from the Electricity Supply Authorities' point of view?

1.4 Research questions

Any research starts off with a number of questions to which relevant answers are sought. The unknown factors that formed the groundwork for this thesis were initially stated as being:

- Which parameters are relevant in determining the hot water usage patterns?
- How does the heated water behave in a domestic hot water cylinder under varying user demand?
- What hardware/software is necessary to enable effective cylinder control, keeping in mind the possible commercial aspect of any future system?
- What form of forecasting model is most suited to giving reliable predictions in the varying situations that will be encountered?
- Can a system learn and adapt to customer water usage patterns, and then modify its electricity purchasing decision to minimise the cost of providing hot water in line with the usage pattern?
- Can an enhanced version of some form of Energy Management System be made capable of forecasting and subsequently utilising the (theoretical) minimum electricity tariffs that change on a per time unit basis?

As the project's timeline progressed these initial indicators were altered to be more specific and, unavoidably, new queries arose. The additional elements of the study can be summarised as follows:

- Analyse water behaviour in the cylinder during heating, standing, and draw off periods.
- Determine the equipment required for automated cylinder control.
- Determine the influence of not one but two heating elements located in different positions on the cylinder.
- Test the effectiveness of the data acquisition software in performing the various tasks.
- Identify the most suitable means of predicting values of discrete temporal series; the series representing patterns of water demand and other (to be determined) relevant parameters.
- Design a Fluid Energy Management System (FEMS) which incorporates, if applicable, the 5 points above, and which utilises the off-peak night tariff rate for heating the expected water demand.
- Obtain data on the amount of usage of hot water in a typical domestic household for subsequent testing of the FEMS.
- Determine a plausible spot price tariff model as is likely to be encountered by a future FEMS.
- Enhance the existing FEMS in order to exploit a future spot price tariff system.

1.5 De-limitations and limitations of the study

As in any real-world research there are constraints on what is available to in the way of assistance; be it budget, time and/or manpower. The limitations initially identified for this thesis were that:

- Only a small number of domestic consumers can provide usage data.
- Forecasting test models and controlling systems will be realised in software.
- A single hot water cylinder is available for modification and research.

1.6 Chapter summary

Having introduced the purpose of the thesis in the present chapter, *Chapter* 2 details the domestic hot water cylinder, the test equipment, the accompanying software and the data acquisition trials. It concludes with a discussion of the results obtained. The significance of being able to forecast the hot water demand under different situations is explained in *Chapter* 3; where an MEM prediction program is developed and tested. It is found wanting, and a feasible alternative is suggested in the form of an artificial neural network.

Chapter 4 then introduces the concept of neural networks and reviews the various configurations available, albeit briefly. The versatility of the neural net for real-world applications is explored in *Chapter* 5 using the networks illustrated in the previous chapter. Some practical issues are noted and the statistical equivalents indicated. Having obtained a good overall picture of the neural network world, the reader moves on to *Chapter* 6 where the focus is on ANNs that are specifically aimed at incorporating the temporal characteristics of a number of data sequences. The most suitable form of neural network for incorporation in the Fluid Energy Management System (FEMS) is selected and its salient features defined.

Chapter 7 reviews energy management and touches on its importance to New Zealand in terms of conservation and efficiency, with a specific view to the role of the hot water cylinder. The concept of energy management systems and their development up to the present day finish this chapter.

Chapter 8 is devoted to the principles and features of FEMS. It explores the hardware and software development in depth and shows the build-up of the input and output data vectors that serve the recurrent neural network. The 'hot water demand' and 'time of use' data collected over a period of 6 months is utilised to train the networks and the subsequent prediction results are closely examined for trend and accuracy. In conclusion a choice is made of the most suitable neural network configurations. In *Chapter* 9 the concept of FEMS is pushed into the realm of energy markets and dynamic tariffs. It is illustrated how an enhanced version of FEMS could take advantage of varying electricity rates in a 24-hour period by predicting a profile of hot water demand coupled to real-time pricing using day-ahead tariffs or on-line tariffs.

Finally *Chapter* 10 re-iterates the ground that has been covered by the thesis and looks at the various avenues of further research that could be undertaken.

In the course of the study a total of three papers were written on the various aspects encountered. The first of these was accepted as a conference paper for the ICNN'95 in Perth, Australia. The two subsequently written articles are under review by various IEEE bodies.

Chapter 2. Water behaviour in a domestic hot water cylinder

2.1 Introduction

If we look at the present form of the hot water cylinder it is evident that the basic design of the cylinder has changed little since its conception sometime in the last century. The changes that have taken place have mainly been in the method of heating; from wood and coal, through oil and gas, to mainly electricity and solar (Doyle, 1990).

A large percentage of the population in industrialised countries around the world use electricity as their sole energy source for heating the required daily amount of hot water. As this water needs to be instantly available, preferably at a high temperature and often in considerable quantities with a high flow-rate, some form of storage container is essential; such as a moderately sized heat-insulated vessel which is capable of meeting this (ir)regular demand.

In a specially commissioned report Hendtlass (1981) states that there are around 1.5 million water heating systems in New Zealand of which about 1.2 million are in houses. The balance are residential-type (less than 500 litres) water heating units that are used in commercial and industrial installations. Many commercial installations have cylinders located in washroom areas, some hotels and motels have installations in each accommodation unit, and factories have cylinders located in numerous places. This number of domestic hot water systems will only have increased in the intervening period given that the population of New Zealand has grown from 3.0 to approximately 3.5 million inhabitants (Encarta, 1997).

In the New Zealand domestic situation the vessel takes the form of a copper cylinder with a dome shaped top and bottom. The cylinder usually holds 180 litres of water at temperatures that can range typically from 10 to 85^{0}C.

The domestic hot water cylinder's basic function is to store water at an elevated, predetermined temperature, so that a volume of water can be drawn off for immediate use. The hot water used is then replaced in the cylinder by cold water and a heating element, controlled by a simple thermostat, maintains the water at the pre-set temperature (Doyle, 1990).

This system has a number of disadvantages. Once fresh cold water is introduced into the container, the existing hot water layer deteriorates and intermixing occurs, essentially decreasing the available volume of water at a high temperature (Sutherland, 1991). Alternatively, in a well-designed cylinder fitted with baffle plates the hot and cold water layer and do not intermix. If further hot water is withdrawn it may appear that the cylinder is full of hot water, until quite suddenly only tepid or cold water is withdrawn. The reason for this may be that the heating element has had insufficient time to reheat the incoming water or, with a badly positioned sensor, that the thermostat has failed to register a temperature drop.

Another difficulty with the cylinder is that the thermostat and the heating element reheat the cold water as soon as the thermostat registers a temperature drop. It takes no account of the time of the day. Thus the thermostat may be causing the heating element to draw power at a time of day when the power is most expensive and in all probability before the heated water

is actually needed. Also, known thermostats take no account of the hot water requirements or hot water use habits. Thus the heating element may be heating a cylinder full of water ready for immediate use, when the next use of hot water may not be for some eight hours.

Earlier laboratory and computational experiments (Sutherland, 1991) have shown that under the right conditions a tank of water will achieve a high degree of *stratification*, resulting in a sharp, well defined boundary between the hot and the cold zones of water (*Figure 2.1*). In an *ideally* stratified tank the interface between hot and cold water would have *zero* thickness.

Factors such as inlet jet mixing, heat conduction, and side losses can degrade this boundary creating a significant thermocline (The region where both hot and cold water mixes and the temperature gradient varies approximately linearly from the cold zone temperature up to the hot zone temperature). This region needs to be kept to a minimum if the *stored energy efficiency* of the cylinder is to be preserved (Parker, 1993).

Al-Marafie et al. (1989) investigated the influence of tank geometry on thermal stratification. They found that increasing the height to diameter ratio of the tank to 3 or 4:1, significantly increased the effectiveness of the storage tank.

Lavan et al. (1977) experimentally studied thermal stratification in hot water storage systems. In particular, a high extraction rate from plastic cylindrical vessels was emphasised. The effect of inlet and outlet port configuration on thermal stratification was also studied.

Evans (1984) in his computer model determined the degree of stratification present within a standard domestic hot water cylinder and concluded that a high degree of stratification was achieved. He therefore found it suited to off-peak energy storage.

A computer simulation study by Parker (1993), using a hot water demand profile as shown in *Figure 2.1*, concluded that if the water in a cylinder remained stratified, then the water temperature remained acceptable except during the last draw-off.

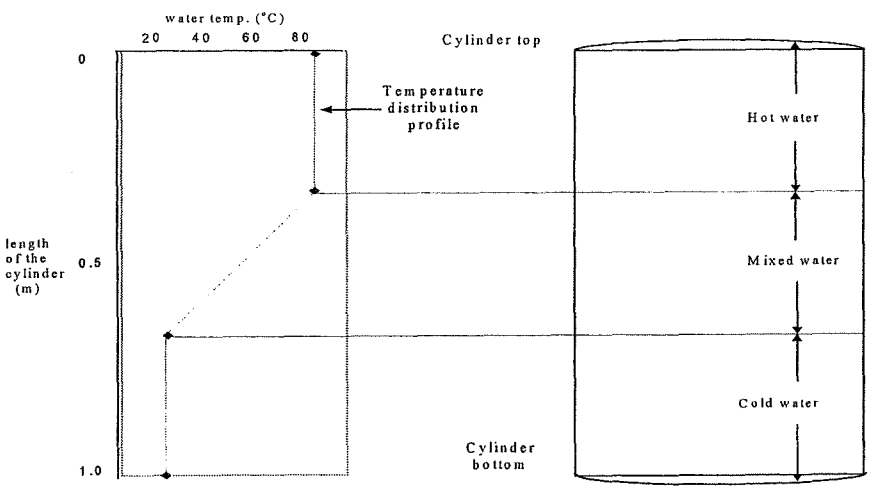


***Figure 2.1* – Typical temperature zones in a hot water cylinder**

However, if some mixing occurred, then the temperature became progressively more unacceptable during the day. Hence the tank design should be such as to discourage mixing. A baffle plate over the cold water inlet to reduce the effect of mixing was suggested. Using the same demand profile it was noted that with a thermostat setting of 50°C. the available water quantity was inadequate and the temperature became unacceptable during the third from last draw-off. Increasing the setting to 70°C made the temperature acceptable and the

required quantity was almost provided. With a setting of 80°C, the quantity was satisfied but there was a potential hazard from scalding (ACC, 1990).

Heat conduction down the cylinder walls (short circuit effect) and through the water, degrades thermal stratification. Al-Marafie et al. have shown that tank geometry has a significant effect on thermal stratification. In cylindrical storage tanks with small H/d (height/diameter) values; the short circuit effect and heat conduction through the fluid, become more significant. In tall tanks with large H/d values; the resistance to heat loss through the fluid and the tank walls becomes greater. With larger H/d values; however, greater heat loss to the environment occurs because of increased tank surface area. A H/d value of between 3 and 4 was found to be the optimum tank geometry.

If the tank walls are highly conductive, as in the copper walls of a domestic cylinder, then the walls tend to the average temperature of the tank faster than the fluid; this causes convection currents that degrade stratification.

Stratification degradation can also be caused by heat retention in the tank walls when the thermocline moves up the tank. The temperature of the tank walls, because of their finite thermal mass must lag behind the water temperature. Water close to the tank wall will get heated, producing convection currents. Calculations by Lavan et al. (1977) indicate that the quantity of heat conducted through the tank walls is considerably larger than that lost through the tank insulation, or through heat transfer by conduction between the hot and cold water volumes. Interestingly, it has been found that a tank with no insulation maintained thermal stratification better than a tank with insulation placed outside of conductive supporting walls. The reason being that a tank with no insulation has a heat leak to the room so that axial conduction in the wall does not play an important role in the heat transfer processes within the tank. In the above experiment however, the tank walls were 14mm thick. The walls of a domestic hot water cylinder are quite thin, typically less than 1mm. The thin wall thickness of the tank will help to minimise effects such as the short circuit effect.

Sutherland (1991) concludes that the domestic hot water cylinder appears to satisfy most of the requirements necessary to attain a high degree of thermal stratification. It has a height to diameter ratio of 3.1:1 and the inlet temperature of the incoming water is lower than the lowest temperature within the tank. The thin walls and good insulation will help to minimise short-circuiting and side losses.

Sliwinski et al. (1978) investigated stratification in thermal storage during charging. The tank in this experiment had a side inlet and low volume flow rates were used; the result was good stratification. When different inlet port configurations were used (i.e. bottom inlet) an increase in turbulence was noted, upsetting the layering effect.

Zurigat et al. (1988) also experimentally investigated the influence different inlet configurations had on a stratified tank. Their apparatus used fresh and saline water to produce a density gradient rather than hot and cold water.

Both of these studies used a parameter called the *Richardson number* to define the degree of stratification. The gradient form of Richardson number represents the relative importance of inertia and buoyancy forces. Both of these studies also observed the same relationship between the size of the inlet jet mixing region and Richardson number. They found that there exists a *critical* Richardson number below which the volume of the mixing region within the cylinder rises sharply. Increasing the Richardson number to much greater values than the critical Richardson number was found to have little effect on the size of the mixing region.

Richardson number is given by:

$$R_i = \frac{\Delta\rho\, g\, H}{\rho_m v^2} \tag{2.1}$$

where
- g = acceleration due to gravity (m/s^2)
- ρ_m = mean density (kg/m^3)
- $\Delta\rho$ = density difference between hot and cold water (kg/m^3)
- v = velocity of water at the tank inlet (m/s)
- H = tank height (m)

An earlier study by Parker (1982) used a computer simulation to show that the basic consumption in a domestic hot water system was increased by between 17 and 37% because of the losses in the system pipe-work, dependant on its arrangement. The study also confirmed that as the thermostat setting was altered, so the quantity of hot water used varied *inversely* (Parker et al., 1991).

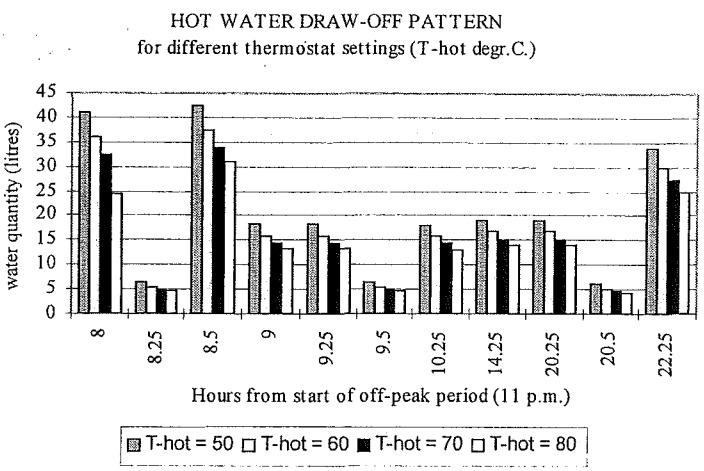


Figure 2.2 – **Typical hot water demand profile for a family of four persons.**

2.2 Objectives

Only limited information could be found in existing literature pertaining to hot water usage, its behaviour when subjected to intermittent draw-off and its management in domestic cylinders. Apart from preliminary project work by third year engineering students at the university of Canterbury most of the existing books and papers concentrated on the water stratification, the theoretical aspects, computer simulations, losses in distributing hot water around the house, conserving energy or applying solar-heating (Lefas, 1987; Pitkin, 1979). It was thus deemed of importance to establish what actually eventuates inside a hot water cylinder when water is heated, drawn-off, left standing, and eventually re-heated.

Before embarking on a series of tests the following objectives were established:

- Obtain a good basic understanding of past and recent discoveries and advances in this field.

- Decide which parameters will possibly influence cylinder heating.
- Determine the necessary data acquisition and control equipment. And then design the required apparatus or justify the purchase.
- Interface the different equipment in order to build an automated and fully instrumented test rig.
- Write a software program that carries out the data acquisition and controls the cylinder and associated peripheral equipment.
- Perform tests on a 180 litre hot water cylinder to determine the affect of different parameters on the cylinder's thermal behaviour and on the various sensors, i.e. the influence of water mixing, draw-off volume, element heating rate, sensor accuracy, effective number of sensors etc.
- Incorporate the data in a new, software based, Energy Management System (EMS) and establish the methods for controlling and predicting hot water usage in a domestic environment.

2.3 Experimental hardware

A complete schematic of the experimental hardware used in the cylinder trials is shown in *Figure 2.12*. The individual items in the set-up are discussed below.

2.3.1 The hot water cylinder

The trials were carried out with a standard 180 litre, medium pressure, domestic hot water cylinder as provided by Multi-machinery in Christchurch. The cylinder consists of a copper pressure vessel surrounded by 50mm (injected) foam insulation and galvanised tin outer cladding. Dimensions of the copper tank are approximately 1.0 m high by 0.46m diameter (*Figure 2.3*). Detailed specifications are given in *Appendix* G.

Exchanging the inlet *pressure-reducing* valve allowed the cylinder to be operated as either a *medium* pressure (75kPa) or *high* pressure (100kPa) vessel. A baffle plate located on the inside, near the inlet, minimises water disturbance by forcing the in-rushing water down to the bottom of the cylinder[1].

The cylinder did have one feature not normally found on domestic cylinders in New Zealand and that was a *second* 3kW heating element located 1/3 of the way down the length of the cylinder. This was in addition to the 3 kW element in the classical position at the bottom of the tank.

The controlling software switched the elements off when the temperature of a specified volume of water reached a preset level. To measure the temperature at various heights in the cylinder a thermal sensor strip (*Section* 2.3.2) was located on the outside of the tank and was sandwiched between 2 thin layers of foam. It was possible to access the strip via a removable panel.

[1] A possible disadvantage of this is that the strong flow could disturb any sediment deposits on the bottom of the cylinder. As the cylinder was new and therefore had no deposits this could not be verified.

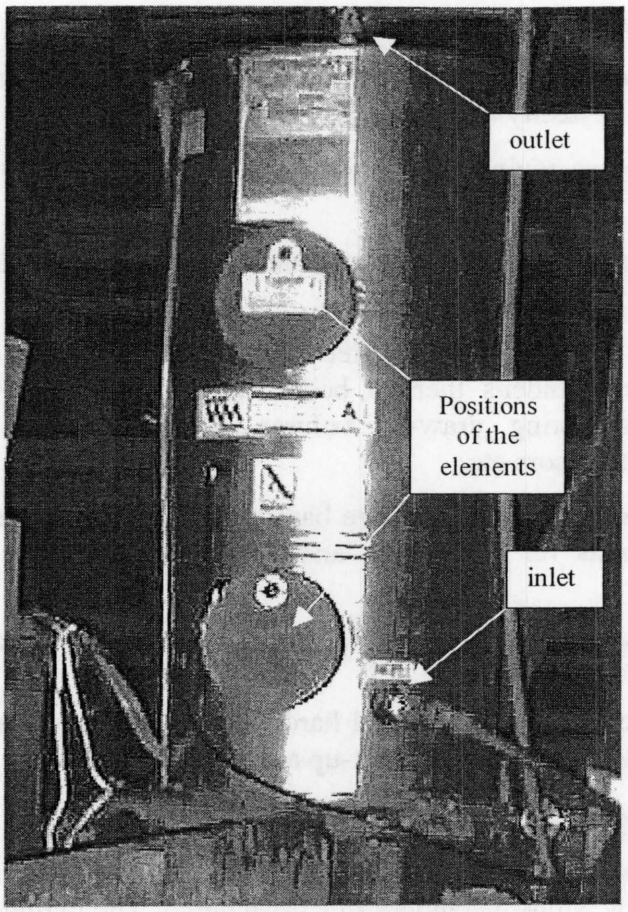


Figure 2.3 – **A domestic 180 litre hot water cylinder installed in an attic.**

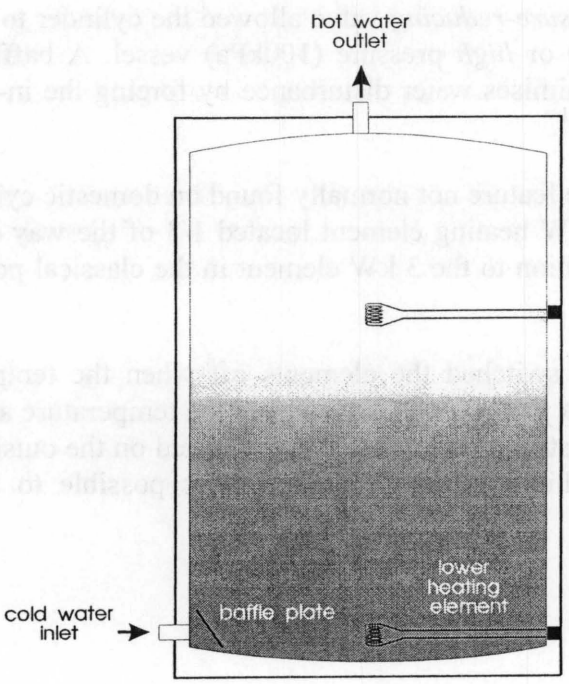


Figure 2.4 –**Sectional view of the 180L hot water cylinder as used in the trials.**

2.3.2 The thermistor sensors

The 19 Negative Temperature Coefficient (NTC) thermistors that made up the sensor strip were allowed to contact the cylinder wall by means of an equal number of holes punched into one side of a foam strip and were mounted 50 mm apart from each other. They were designated "Th" and numbered from 0 to 18, with Th0 being situated at the top of the cylinder (*Figure 2.5*).

The thermistors, each with a resistance of 1000 ohms at 25^0C (R_{25}= 1kΩ), have a typical dissipation factor of 8.5mW/^{0}K and thus require a small amperage constant current source to give reliable readings (Horowitz, 1981). For this purpose a purposely-designed 0.2mA source was constructed which ensured that I^2R heating did not exceed 1mW.

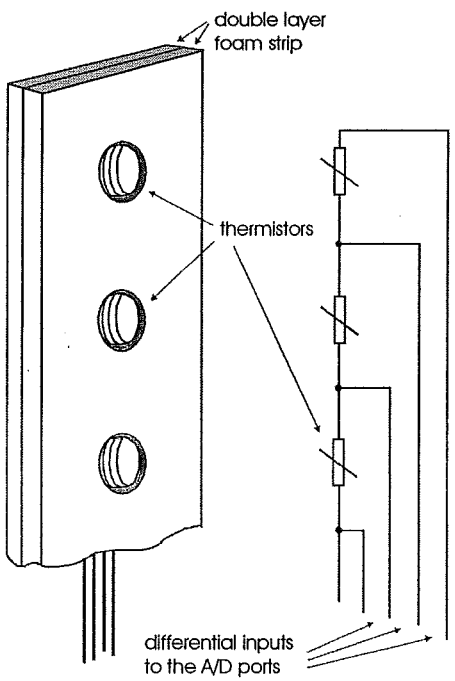


***Figure 2.5* – a section of the thermistor "strip"**

2.3.3 The acquisition/control system

The data logger and automated control system took the form of a 486DX/33 IBM compatible computer fitted with two (16 bit, ISA slot) Advantech PCL-814 multi-purpose DAS cards (Advantech, 1992).

2.3.4 The digital counter

A digital counter had been assembled which allowed the output from the flowmeter to be read by the digital input (DI) of the DAS card. Two CMOS 74HCT163 integrated circuits have been used. Design of the counter was not quite as straightforward as the simple circuit (see *Appendix* H) would lead one to believe. Signal condition was necessary on the gate inputs of the ICs to avoid the indiscriminate counting of the additional spurious spikes, eventuating from switch bounce in the reed relay; the relay being actuated by a small magnet located on the rotating shaft of the flowmeter.

The counter was fitted with eight LED's for visual verification of the signal status on an equivalent number of signal output lines. These lines subsequently fed the input on eight DI

channels on the PCL-814 card. The software periodically sampled the status of these eight channels.

2.3.5 The heating elements and on/off valves

The two 3 kW heating elements and five small motorised on/off valves (Motortrol valves from Erie Manufacturing Co. (Canada) Ltd.) were DO operated by a series of 240V, 3A relays. In the case of the heating elements the relays acted in an auxiliary capacity and activated two 480V, 40A relays capable of handling the larger current demands.

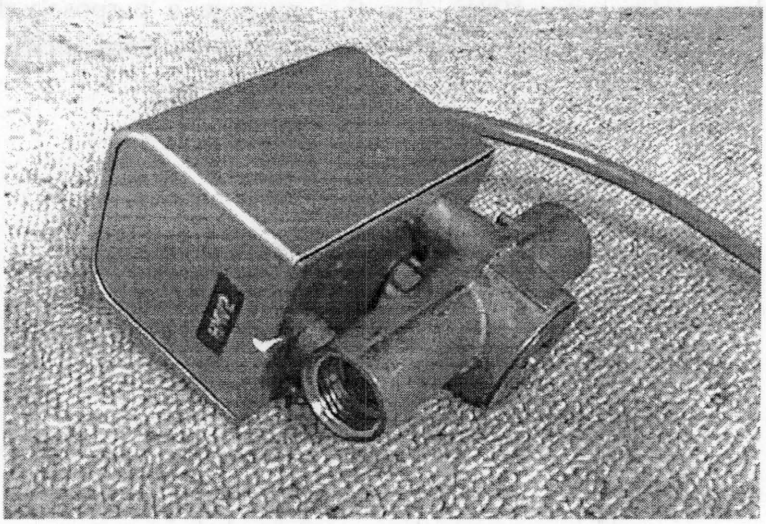


Figure 2.6 – **Motorised on/off valve**

Four of the five on/off valves were situated in the cold water input line feeding into the side of the cylinder. As there was a need to be able to vary the flow each of these four valves was preceded by a manually operated on/off valve. These valves then were individually calibrated to give flowrates of 1, 2, 4, and 8 litres per minute. Combinations of these settings give a range of 1 to 15 litres per minute.

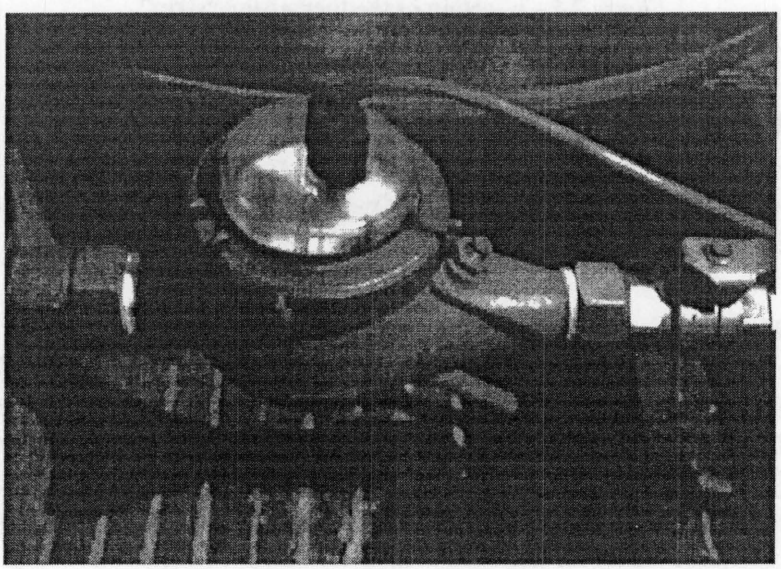


Figure 2.7 – **The flowmeter with the reed relay probe inserted in the top cover.**

The fifth valve was mounted on the hot water outlet line exiting from the top of cylinder. This line went straight to the drain.

2.3.6 The flowmeter

The flowmeter, also situated in the cold water input line, measured litres throughput and provided the quantitative input (1 pulse per litre) to the digital counter by means of a small magnetically actuated reed relay (*Figure 2.7*).

2.3.7 The constant current source

The voltage across each thermistor in the foam strip needs to be directly proportional to *only* the resistance of the thermistor. It should not be dependent on the current flowing through the thermistor, which should therefore be a known *constant* value. Hence the use of a constant current source that in this case was based on a design incorporating an LM317 three-terminal adjustable regulator (Horowitz and Hill, 1989).

However, an added complication is the fact that *self-heating* in the thermistors needs to be avoided at all costs. The *dissipation factor* for the 1kΩ thermistor is 8.5 mW/°K, meaning that I^2R heating should be kept below 8.5 mW if the reading accuracy is not to suffer. As a result the constant current needs to be small, typically in the order of mA's; the simplified circuit in *Figure 2.8* achieves this by adding an op-amp follower to the LM317 regulator *(see Appendix* H *for a detailed circuit diagram).*

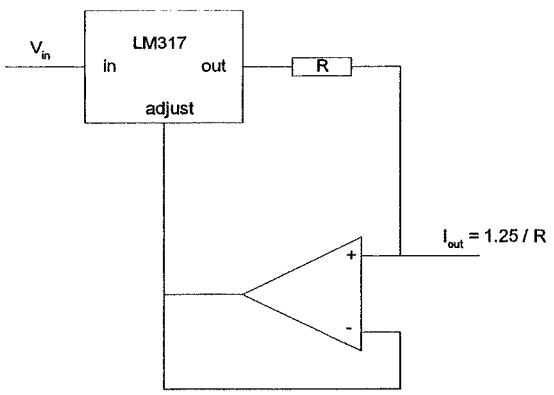


***Figure 2.8* – Block diagram for a small current source.**

2.3.8 The PCL-814 DAS card

The PCL-814 card is designed to be used in industrial and laboratory data acquisition environments as a multi-purpose DAS card for digital input/output (DI/O), counter/timer, D/A, and A/D applications (*Figure 2.9*). It provides up to 16 channels of differential analog signal inputs and 16-bit I/O connectors that allow various on/off process control and monitoring applications. The analog input ranges were software programmable, as was the selection of DMA channel and IRQ level.

The cards for this project were fitted with 14-bit A/D, 100 kHz modules; one to each card (*Figure 2.10*). Features include:

- 14-bit A/D conversion.
- 100 kHz sampling rate.
- Pacer trigger for A/D synchronisation.

- Software programmable analog input range.
- Programmable DMA channels and IRQ levels.
- On-board auto channel scanning circuitry.
- 'C' language software driver.

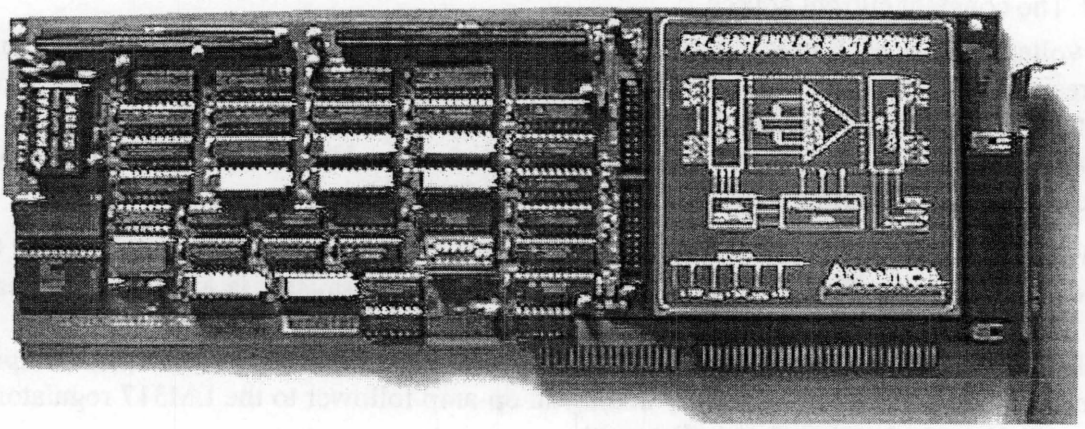


Figure 2.9 – **The PCL-814 data acquisition card (A/D and I/O).**

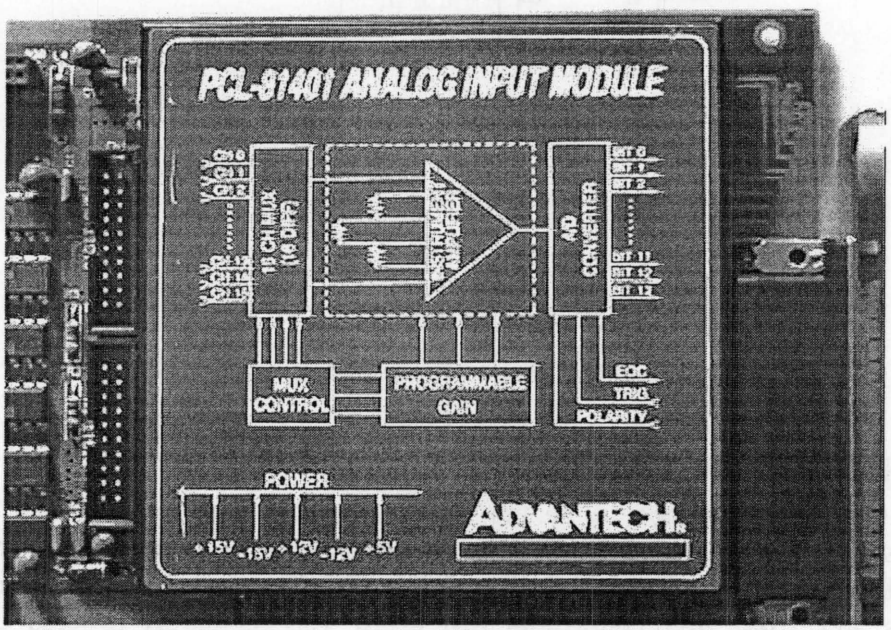


Figure 2.10 – **Close up view of the A/D module as fitted to the PCL-814.**

2.4 Data acquisition and control software

2.4.1 The test program

Existing data-acquisition software packages were found to be incompatible with the new PCL-814 boards and all were geared towards simple data acquisition and manipulation. They

lacked suitable real-world interaction (DI/DO's) and none had standard control functions allowing the user to easily direct peripheral equipment.

A considerable amount of time was invested in writing a comprehensive test program, which was flexible enough to be easily modified, as well as made use of the board-manufacturer supplied drivers. Yet it had to be robust enough to drive a complete test cycle and operate a collection of 240 VAC mains powered hardware; all without operator intervention.

The final Controlling and Data Acquisition software consisted of seventeen interacting modules written in the language 'C', which drove the entire system and collected the data for a complete test-cycle without operator interference. The decision flowchart for the software is shown *Figure 2.11*. Additional information on the software can be found in *Appendix* I.

2.4.2 The PCL- 814 driver

The driver supplied with the PCL-814 card is a "terminate-and-stay-resident" (TSR) program that operated in the background during the execution of the data acquisition and control program. In the course of the writing the software, the driver had to be modified to optimise the use of software triggered A/D conversions.

The driver performs by means of so-called "function calls" whose purpose it is to initialise the card, set the input range, set the start and stop scanning channels, perform the A/D conversions, etc.

2.4.3 PCL- 814 calibration

A calibration utility program was provided with the PCL-814 card. The program recorded the A/D module's total *offset count* when the differential input is at 0 volts. Adjustments are accomplished by using the following equation:

$$\text{Actual count} = \text{Measured count} - \text{Offset count} \qquad (2.1)$$

2.4.4 Data collection

Part of the information gathered underwent data processing before being stored directly into comprehensive files. Each file gives details of the measured parameters and elapsed time.

As a precaution the *raw* data collected was also stored in separate files as a form of backup and, more importantly, allowed for corrective or alternative processing if this was ever required.

2.5 Practical considerations

Domestic hot water use tends to be intermittent rather than continuous. The time elapsed between draw-offs can be considerable. For example; a bath or dishwasher will be run, with perhaps several hours before the next usage.

With this in mind the trials also allowed for these intermittent conditions; for practical purposes the time-scale was shortened with due time being given for the water circulation, if present, to settle between draw-offs.

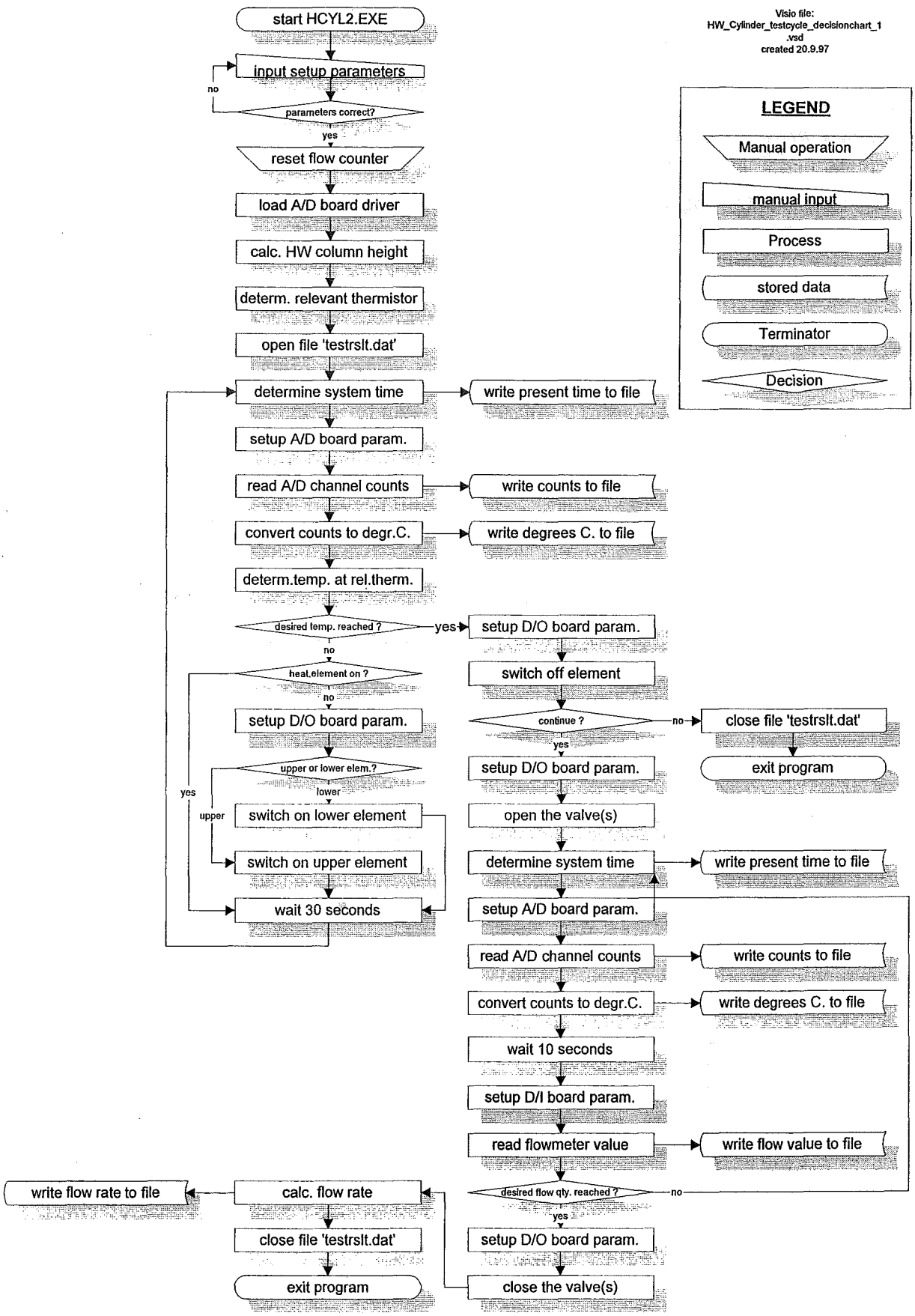


Figure 2.11 – **Decision chart for the control and data acquisition software.**

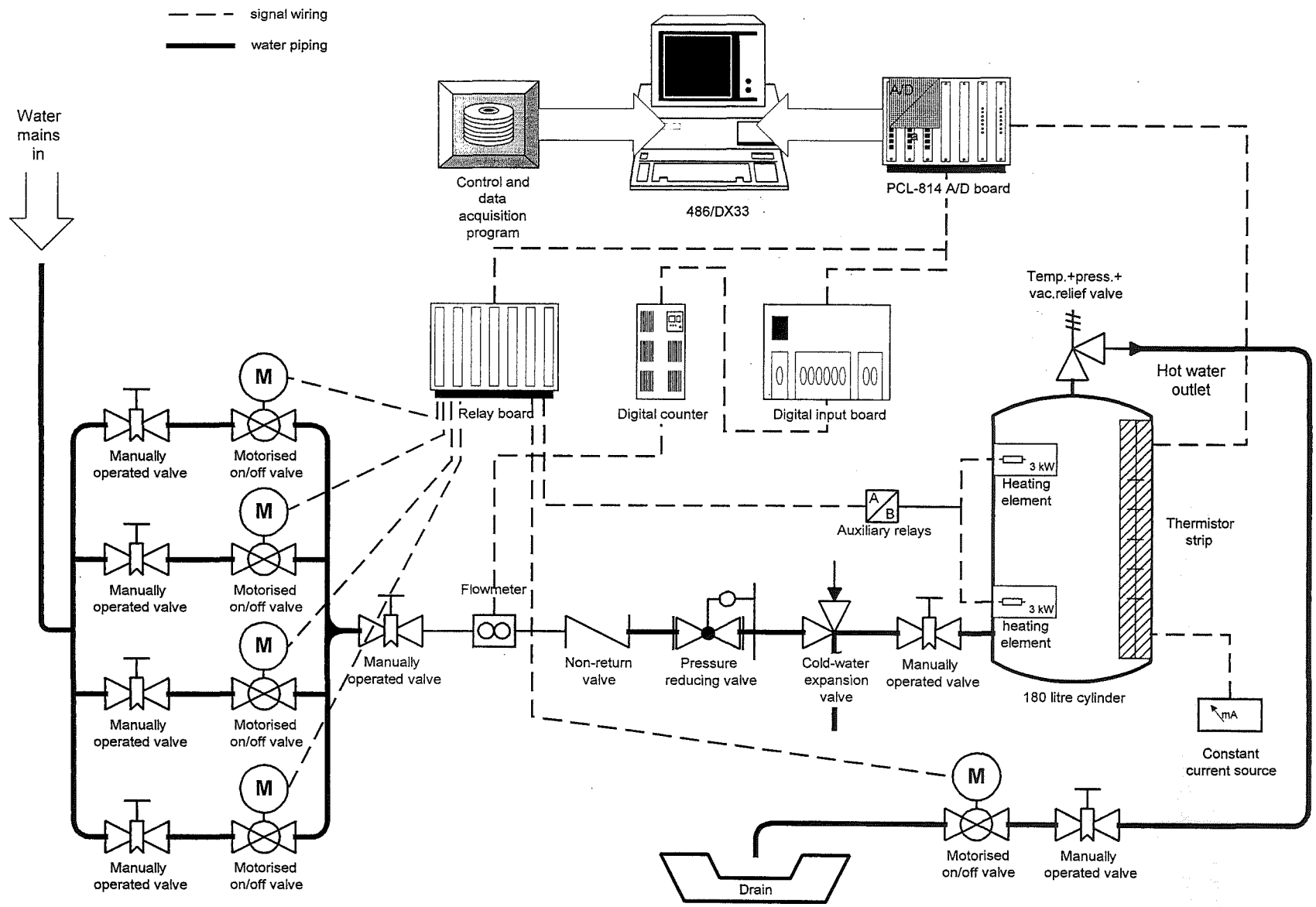


Figure 2.12 - **Schematic of the experimental equipment.**

In domestic households the draw-off flow rate will also vary with use. An average shower for example, draws off water at 6 to 8 litres per minute; the maximum flowrate of 15 litres per minute is representative of a bath or washing machine being filled. In the experimental setup the flow rate was varied by using alternative combinations of the four on/off valves on the inlet.

The amount of water to be removed at each draw-off ranged between 15 litres to 30 litres. Some results are given with 120 litres having been drawn, this represents a sizeable volume of the tank and was considered representative for the water volume of a typical bath or other household activities in a one day period.

The initial hot water temperature is another parameter that was varied during the trial runs. The settings used ranged from 50 to 90°C.

2.6 Method of analysis

For the purpose of analysing results a layer of water 50mm thick was associated with each thermistor. It was assumed that the temperature within each layer remained constant in the radial direction. The temperature at side of the cylinder wall, as measured by a thermistor, was assumed to be the mean temperature of that layer in the axial direction.

2.7 The temperature sensor

2.7.1 Choice of temperature sensor

Four different types of sensors have been considered for application in this project: platinum resistors, thermistors, thermocouples, and semi-conductor integrated circuit sensors.

Platinum resistors are high quality, highly linear sensors used for precision measurements. They are usually employed in a bridge configuration where resistance changes are converted to bridge output voltage changes. Besides their considerable cost, these sensors have a relatively low temperature coefficient so that the bridge output is relatively low. This requires the use of high quality, very low drift operational amplifiers which tend to increase further total system cost and, also significant for a domestic situation, reduce long term stability.

Thermocouples are popular sensors offering a wide measurement range and small thermal capacitance. However, neither of these factors was of major importance for the application considered in this project. Measurement range is fairly limited (e.g. 0°C to 100°C) and the thermal capacitance of the hot water cylinder is many orders of magnitude higher than that of the sensor, so that the thermocouple low thermal capacitance advantage is not utilised. The disadvantages of thermocouples are considered to be their non-linearity, although it is not as severe as that of thermistors, and their low output voltage. Also the thermocouple output is a measure of the temperature difference between the hot and the cold junction so that a special signal conditioning circuit has to be used to convert the thermocouple output to absolute temperature measurements. Such circuits, electronic ice points, use their own absolute temperature sensors. However, their cost offsets the low cost advantage of thermocouples and the use of the additional temperature sensor tends to reduce measurement accuracy. For completeness it should be mentioned that long-term drifts in the signal conditioning circuit also reduces measurement accuracy. Taken as a whole the thermocouple was considered a poor choice for the given application.

Semiconductor IC sensors exploit the temperature dependence of the transistor base-emitter voltage dependence. Through suitable IC manufacture and trimming, highly linear and stable IC sensors are produced. The signal conditioning circuit is fabricated on the same waver resulting in low cost, easily interfaced sensors. Typical IC sensors (e.g. the National LM 35) provide a measurement accuracy of 0.25°C and sensitivity 10mV/°C. The cost of these sensors is in the order of high quality thermistors. The main disadvantage of these sensors is their limited measurement range, which is usually 0°C to 70°C. Recently introduced sensors provide a larger range, typically in the order of -30°C to +130°C but this is accompanied by a significant price increase. Another disadvantage is that IC sensors have a considerable wafer-ambient heat resistance so that considerable self-heating takes place if they are continuously powered. This problem can be overcome by powering the sensor intermittently.

As such the IC based sensor can be considered too expensive and involved for the needs of the project. However, it must be remarked that for a *commercial* system with possibly fewer sensors in conjunction with large bulk-orders will bring the price down to reasonable levels; in those circumstances this type of sensor can be considered the appropriate choice. It is interesting to note that semiconductor circuits in parallel will yield the average sensed temperature, and when in series the minimum sensed temperature.

Thermistors are semi-conductor devices that exhibit a negative coefficient of resistance with temperature, typically in the neighbourhood of -4% per °C. They are available in all sorts of packages, ranging from tiny glass beads to armoured probes. Thermistors intended for accurate temperature measurement typically have a resistance of a few thousand ohms at room temperature, and they are available with tight conformity to standard curves. Their large coefficient of resistance change makes them easy to use, and they are inexpensive and stable. Thermistors are a good choice for temperature measurement and control in the range of -50°C to +300°C. Because of their large resistance change with temperature, thermistors make no great demands on the circuitry that follows.

Thus, despite its non-linear, almost logarithmic response, it is by far the cheapest and easiest to apply for this project. Add to this the fact that a foam strip fitted with an appropriate number of thermistors was readily obtainable and that the test-software could effortlessly compensate for the device's non-linearity, than the result is that the thermistor is the logical choice.

2.7.2 Thermistor temperature sensors

The thermistor functions as both sensor and transducer by converting the non-electrical parameter of temperature to an electrical signal in the form of a voltage. The thermistor is one of the cheaper sensing elements and consists of ceramic material made by sintering mixtures of metal oxides into whatever shape is required, in this instance a disc. The size is quite small which is a distinct advantage.

Thermistors are a form of semiconductor and their resistance depends on their composition. A minority has positive temperature coefficients (PTCs) where resistance increases with increasing temperature. The majority have negative temperature coefficients (NTCs), their resistance falling with increasing temperature. Although the change in resistance is larger than for another popular temperature sensor, platinum (Pt-100), the resistance changes non-linearly with temperature.

The relationship between resistance and temperature for a NTC thermistor is given by:

$$R_t = R_\infty e^{\frac{B}{T}} \tag{2.2}$$

where R_t = thermistor resistance at temperature t °C (Ohms);

R_∞ = resistance that thermistor tends to at high temperatures (Ohms);

B = thermistor material constant (°K);

T = temperature t in Kelvin (°K), where T = t + 273.

There are two methods for tackling the non-linearity of the thermistor, whose resistance versus temperature response is shown in *Figure 2.13*. Either linearise the device over a small temperature range by placing an accurate resistor in series or parallel with the thermistor, or make the appropriate calculations to find the temperature which corresponds to the voltage signal.

The first method is of value when working with a single thermistor in a limited range, typically 0°C to 40°C with an accuracy of 1%. Faced with a greater number of devices, a higher temperature range and having plenty of computing power, the second method[2] was the one opted for. Applying a *constant current* to the thermistor and measuring the voltage across the thermistor allows us to determine the resistance and hence the temperature (see *Section* 2.7.5).

When applied to (2.2) and substituting the known manufacturer's data for the 1 kΩ Negative Temperature Coefficient (NTC) thermistor the following equation can be obtained:

$$R_t = (0.0026)\ e^{\frac{3825}{T}} \qquad (2.3)$$

Thus stated, the 1 kΩ thermistor has an R_∞ of 0.00266 Ω with B equal to 3825°K. *Appendix* A gives a complete overview of the characteristics of the type of thermistor used (Phillips Co., 1984). Typically the field accuracy of the thermistor is ± 1°C, whereas the thermistor element itself has an accuracy of ± 0.2°C. (Levermore, 1992)

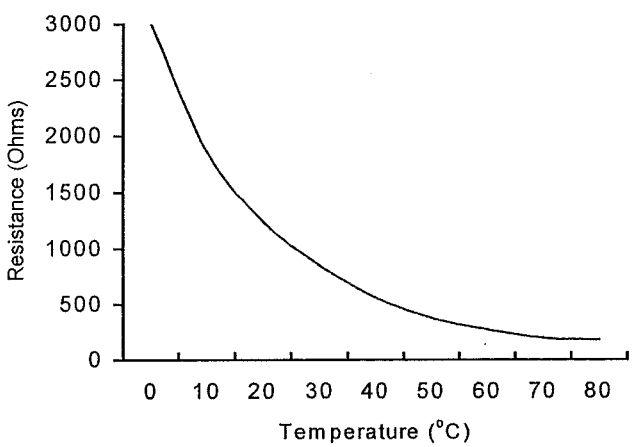


Figure 2.13 **- Negative Temperature Characteristic of the 1KΩ thermistor**

[2] This method can be modified to suit a less powerful and more memory limited microprocessor than the 80486 used for this project. The modification could take the form of a look-up table put in an EPROM and contain an appropriate number of voltage and temperature values.

2.7.3 Thermistor dynamic response

The dynamic response of the thermistor affects how quickly it approaches the measured water temperature. Until the thermistor is in equilibrium with the water there will be some error in the sensed temperature. The thermistor's dynamic response comes from the sensor having its own thermal mass which takes time to heat up and cool down as the water, or more correctly the copper surface of the cylinder, whose temperature is being measured changes.

This is described by the following equation:

Rate of change of heat into sensor	-	Rate of change of heat out of sensor	=	Rate of change of heat stored in sensor

The response of the thermistor sensor is determined by its reaction to a sudden change in temperature of the water it has been measuring for some time and with which it is in equilibrium. This instantaneous step change in the water temperature causes the sensor temperature to change, but as the sensor takes time to heat up or cool down it does not instantaneously achieve the new water temperature. The larger the mass of the sensor, the larger the time constant and the slower the response. The copper surface and the foam mounting strip surroundings also add to the mass.

The time constant can be measured as the time for the sensor temperature to rise to 63.2% of its full temperature rise. Typical sensor time constants have been taken from Levermore (1992) and are shown in *Table 2.1*.

Sensor element	Time constant (min)
Thermistor in water	0.2
Thermistor in pocket in water	0.6
Thermistor in still air	2.4

***Table 2.1* – Typical sensor Time Constants**

Measurements made with the thermistors (installed in the foam strip) in still air returned an average value for the time constant of 1.0 ± 0.2 minutes (*Table 2.2*). Due to practical limitations these measurements could not so easily be repeated with the strip installed against the cylinder wall; as previously stated this is made of copper that presents a rather large thermal mass. Add to this the fact that the foam strip and cylinder jacket act as effective insulators and it would not be unreasonable to assume that the time constant should be a larger value for a thermistor installed against the cylinder.

A rather crude method was devised to glean some idea of what the time constant would be with the thermistors installed on the cylinder. The technique consisted of rapidly draining a cylinder full of hot water and subsequently measuring how long it took for the bottom-most thermistor to arrive at the (known) temperature value of the replacing cold water. The value arrived at was 5.0 ± 0.5 minutes.

2.7.4 Improving thermistor surface contact

As the thermistor sensors are not in direct contact with the water they are measuring the outside surface temperature of the cylinder wall, whereas the inside surface of the wall is in

contact with the water. However as the surface is made of copper, a material with good conductivity, the *thermal resistance* will be small (Rogers et al., 1967).

Sensor element	Time constant (min)
Thermistor in still air (with strip)	1.0
Thermistor against cylinder (w.strip)	5.0

***Table 2.2* – Time constants derived for the 1 kΩ Philips thermistor.**

Unfortunately, there will be more resistance between the surface of the sensor and the outside of the cylinder. This is due to the two surfaces being rough on a microscopic scale and making contact at relatively few points of a small area. *Figure 2.14* illustrates this contact between two surfaces.

***Figure 2.14* – Two 'smooth' surfaces in contact on a microscopic scale.**

Where contact is made, the area is often small and it forms a *constriction resistance* to heat flow, like three lanes of traffic being funnelled in to one lane, with the resultant hold up or resistance to flow (Levermore, 1992). Levermore further states that there will be convection and radiation heat transfer across the gaps between the surfaces, but it is assumed that the resistance of these two modes is much greater than that of the constriction resistance.

To reduce the constriction resistance the area of contact needs to be increased. This can be achieved my putting some silicon grease or oil into the pocket where the sensor is located, to fill in the gaps and so increase the area of contact. The conductivity of the fluid is much larger than the air it replaces and it therefore shorts out the constriction resistances. A disadvantage here is that the oil or grease can only be retained in the pocket if this is positioned vertically. An alternative is to ensure that adequate pressure is exerted on the thermistor tip to obtain good contact with the surface of the object whose temperature is to be measured. The influence of heat loss due to radiation and convection needs to be minimised by good insulation around the area of measurement. If these factors are ignored than Levermore concurs that errors of 20% or more below the fluid temperature can be made. The actual effect is difficult to quantify as little work has been done to measure thermal constriction resistances.

Both the requirements of good contact and good insulation were well-satisfied with the test-equipment used, by forcing the thermistor strip against the tank's copper surface and surrounding it with foam insulation before replacing the outer steel cladding. Furthermore, obtaining an absolutely accurate value for the temperature was less important than obtaining

a *consistent* temperature reading; and especially the temperature difference between the individual thermistors was relevant. In other words; it was more important to get a good intimation of the temperature (within ± 2°C), and to be able to determine the boundaries of the layers of water with varying temperatures.

Although it is a means not employed in this study, it is interesting to mention that filling the thermistor pocket with silicone grease could have enhanced thermistor-to-copper surface contact further. The pocket boundaries in this case being formed by the foam strip and the copper cylinder surface.

2.7.5 Thermistor data conversion

The PCL-814 A/D card does not naturally output a temperature value for a given voltage reading. In fact, the card's circuitry converts the measured voltage to an equivalent 'count' value. Prior to allowing any input the card is calibrated by grounding the input ports and running a calibration program designated 'CAL814.EXE' (see *Section* 2.4.3); the result is stored in a configuration file 'PCL-814.CFG'.

The PCL-814 allows the user to choose from a number of voltage input ranges. Given the fact that the voltage drop across each thermistor is small due to the low constant current (thus avoiding self heating problems); it is clear that the smallest voltage input range will afford the best resolution. The maximum count value the A/D card can output is 16383, regardless of the input range selected.

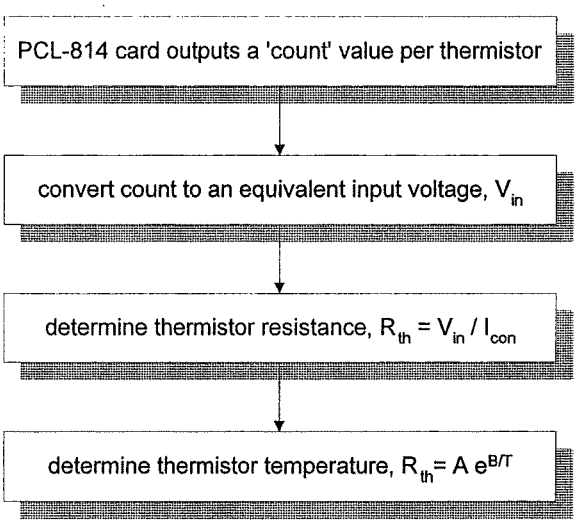


***Figure 2.15* - Conversion stages from A/D input reading to final temperature value.**

Figure 2.15 shows the conversion steps that need to be taken in order to arrive at a legitimate temperature value, as sensed by a thermistor on the foam strip fitted to the cylinder.

With a chosen range of 0 to 1.25 V, the number of counts per volt is,

$$C_v = \frac{\text{maximum count}}{\text{maximum voltage}}$$

$$= \frac{16383}{1.25}$$

$$= 13106.4 \text{ counts/volt} \qquad (2.4)$$

The voltage across an individual thermistor Th, is represented by V_{in} and can be determined by knowing the count associated with it:

$$V_{in} = \frac{count}{C_v}$$

$$= \frac{count}{13106.4} \quad (2.5)$$

Having established V_{in}, the actual thermistor resistance is:

$$R_{Th} = \frac{V_{in}}{I_{con}} \quad (2.6)$$

where I_{con} is the known current as supplied by the constant current source.

The final step uses (2.2) and (2.3) to determine the value of the temperature as sensed by the relevant 1 kΩ thermistor:

$$R_{Th} = (0.00266)\ e^{\frac{3825}{T}}$$

$$e^{\frac{3825}{T}} = \frac{R_{Th}}{(0.00266)} \quad (2.7)$$

As the Natural Logarithm of e^{α} is equal to α, (2.7) can be expressed as:

$$\frac{3825}{T} = \ln\left(\frac{R_{Th}}{(0.00266)}\right)$$

$$T_{Kelvin} = \frac{3825}{\ln\left(\frac{R_{Th}}{(0.00266)}\right)} \quad (2.8)$$

This then is the temperature of the thermistor in degrees Kelvin (°K). In the more relevant temperature of degrees Celsius (°C) the equation (2.8) becomes:

$$T_{Celsius} = \frac{3825}{\ln\left(\frac{R_{Th}}{(0.00266)}\right)} - 273 \quad (2.9)$$

2.8 Experimental procedure

At the start of each test-cycle the software requests user input for hot-water quantity, temperature, flowrate, draw-off quantity, scanning period, and allows automatic or manual selection of the heating element.

Each test-cycle was split into two phases. The program would determine which thermistor on the temperature sensor strip reflected the optimum hot water column height (quantity) and in the heat-phase would switch on the most suitably situated heating element. The water in the cylinder would heat until the selected thermistor registered the desired water temperature.

At this stage the software proceeds to the second phase, the flow-phase, of the test-cycle. It switches off the heating element, starts the timer, opens the requisite number of cold water

inlet valves to reflect the desired flowrate, opens the single hot water outlet valve and starts to read the flowmeter input.

Default software settings mean that the temperature sensors are scanned once every 60 seconds in the heat phase and every 10 seconds in the flow-phase; for the latter the scan includes the flowmeter.

The temperature read by each thermistor as well as the quantity of water drawn off is displayed continuously on a computer monitor and is refreshed once every scan period. When the desired quantity of hot water has been drawn off the valves are closed and scanning can be continued if it is deemed essential to obtain a log of the settlement of the remaining hot water in the tank.

Part of the information thus gathered underwent data processing before being stored directly into comprehensive files. Each file gives full account of measured parameters and time.

As a precaution the raw data collected was also stored in separate files as a form of backup and, more importantly, allows for corrective or alternative processing if this is ever called for.

2.8.1 Parameters

In the tests the following parameters were varied and/or monitored:

Heated water quantity (60, 180 litres)

At the start of every trial a fixed quantity of water was heated to the desired temperature. As the cylinder content could only be heated by either the upper element or the lower element, the quantity options were limited to two values: 180 litres (using the lower element) or 60 litres (using the upper element).

Flow rate (4, 8, 15 litres / minute)

In domestic households the draw-off flow rate will also vary with use. An average shower for example, draws off water at 4 to 7 litres per minute; the maximum flowrate of 15 litres per minute is representative of a bath or perhaps a washing-machine being filled. Three flowrates are used to present the results; 4, 8 and 15 litres per minute.

Draw-off Quantity (15, 30 litres)

The quantity of water removed at each draw-off was 30 litres at a time for trials performed with the complete cylinder water content (180 litres) having been initially heated with the bottom element, and 15 litres at a time with only one third (60 litres) of the cylinder water having been initially heated with the top element.

After each 30 or 15 litre draw-off a fifteen minute settlement period was allowed before starting the next draw-off. Immediately prior to each draw-off the temperature distribution in the cylinder was recorded.

This process was continued until all hot water had been drained.

Temperature (50, 65, 80, 90 ^{0}C)

Four different temperature settings were used during the tests. The software 'thermostat' value was set to 50, 65, 80 or 90 ^{0}C at the start of each trial and switched off the lower element when the total volume had reached the requisite temperature. The software

performed the same task in the upper element trials when only a 1/3 of the total volume was heated.

Cylinder water storage pressure (75, 100 kPa)

A complete set of the same tests with varying water quantities, flowrates, temperatures, etc. was performed with the cylinder contents stored at two different pressures; the "medium pressure", being 75 kPa, and "high pressure", at 100 kPa.

For completeness the following tests were also performed, albeit in a more cursory manner:

Re-heating

What happens to a remaining layer of hot water at a given temperature, situated at the top of the cylinder, when the heating element is once again turned on to heat the contents? Does it remain in place, start to increase in temperature or does it disperse? This situation is encountered each night by a large number of New Zealanders who solely use the cheaper electricity night rate between 11 pm and 7 am for heating their cylinder water, storing it for subsequent use during the day and evening time.

Heat dispersal / standing losses

The rate of internal heat dispersal and standing losses to ambient with the given cylinder insulation, with or without cold water being present in the cylinder, was monitored.

Intermittent/continuous draw-off

Domestic hot water use tends to be intermittent rather than continuous. The time elapsed between draw-offs can be considerable. For example; a bath or dishwasher will be run, with perhaps several hours before the next usage. With this in mind the trials also allowed for these intermittent conditions; for practical purposes the time-scale was shortened with due time being given for water circulation, if present, to settle between draw-offs.

2.9 Results

Of specific interest in these tests was how varying conditions in flowrate, initial water temperature, etc. affected the volume of useful hot water available to the domestic user. For our purposes any water with a temperature greater than 40^0C was deemed "useful" warm water suitable for domestic use. Maintaining the integrity of the piston flow region, and to a lesser extent the thermocline region benefits this volume of warm water.

The results are presented in relation to how four different volumes of water, namely, higher than 40^0C (>40degr.C.), piston flow, thermocline and complete mixing behave.

Figure 2.16 shows an idealised temperature profile down the length of the 1 metre tall storage cylinder at some moment in time after a significant quantity of hot water has been drawn off. Note that the initial thermostat setting has allowed the water to heat to 80^0C.

To try and present comprehensible results and clearly illustrate the different effects of flowrate, temperature and pressure on the four volume of interest, we let *Table 2.3* show a snapshot of the situation inside the medium and high pressure cylinders when half (90 litres) of the total volume of the cylinder's hot water has been drawn off. *Table 2.4* show the same data again but than as obtained for when two-thirds (120 litres) of the hot water volume has been replaced by cold water.

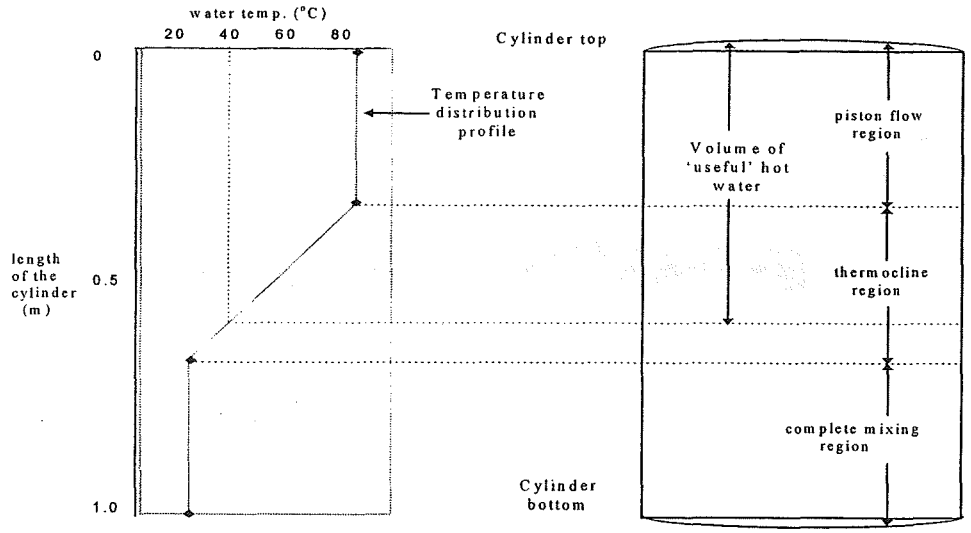


Figure 2.16 -**Regions of interest in a hot water cylinder**

a) A complete mixing region, that increases as hot water is extracted.
b) A thermocline region, that increases with time.
c) A piston flow region, that decreases as hot water is extracted.
d) A 'useful' hot water region, any water at 40°C or higher.

flowrate 4 litres/minute

	50 degr.C.	*65 degr.C.*	*80 degr.C.*	*90 degr.C.*
> 40 degr.C.	43%	49%	56%	58%
piston flow	27%	27%	28%	27%
thermocline	41%	44%	45%	46%
compl.mix.	32%	29%	27%	27%

flowrate 8 litres/minute

	50 degr.C.	*65 degr.C.*	*80 degr.C.*	*90 degr.C.*
> 40 degr.C.	40%	46%	53%	56%
piston flow	25%	27%	27%	27%
thermocline	43%	44%	45%	46%
compl.mix.	32%	29%	27%	26%

flowrate 15 litres/minute

	50 degr.C.	*65 degr.C.*	*80 degr.C.*	*90 degr.C.*
> 40 degr.C.	38%	46%	51%	55%
piston flow	23%	25%	25%	27%
thermocline	44%	45%	46%	46%
compl.mix.	34%	30%	29%	27%

Table 2.3 -**Typical Medium and High pressure data, 90 litres drawn off.**

The four volumes are expressed as percentages; with piston flow + thermocline + complete mixing region quantities together representing 100% of the cylinder volume.

The data from *Table 2.4* specifically have also been plotted in *Figure 2.18* and *Figure 2.19*. These graphs will be used as a reference for presenting the results.

The choice of showing the cylinder's internal status at 120 litre volume draw-off has two reasons; the first is that it is only when approximately half the volume has been drawn off that all three regions of piston flow, thermocline and complete mixing are clearly definable, and second that this volume is on average the amount of hot water drawn off between 6:45 and 9:15 am in a New Zealand household which simultaneously constitutes the largest consolidated removal of hot water in any 24 hour period (Hendtlass, 1981).

flowrate 4 litres/minute

	50 degr.C.	*65 degr.C.*	*80 degr.C.*	*90 degr.C.*
> 40 degr.C.	26%	33%	39%	42%
piston flow	11%	11%	11%	11%
thermocline	45%	50%	53%	53%
compl.mix.	44%	39%	36%	36%

flowrate 8 litres/minute

	50 degr.C.	*65 degr.C.*	*80 degr.C.*	*90 degr.C.*
> 40 degr.C.	24%	30%	36%	39%
piston flow	9%	11%	11%	11%
thermocline	47%	50%	53%	53%
compl.mix.	44%	39%	36%	36%

flowrate 15 litres/minute

	50 degr.C.	*65 degr.C.*	*80 degr.C.*	*90 degr.C.*
> 40 degr.C.	21%	27%	35%	39%
piston flow	5%	7%	9%	11%
thermocline	50%	54%	55%	54%
compl.mix.	45%	39%	36%	36%

Table 2.4 - **Typical Medium and High pressure data, 120 litres drawn off.**

2.9.1 Cylinder Water Pressure

It was expected that the high pressure cylinder would suffer adversely in maintaining it's piston flow volume due to increased turbulence caused by the higher pressure jet of the in-flowing cold water which replaces any hot water that is drawn off. The full range of tests clearly indicates that for a standard cylinder, fitted with a baffle plate, increasing the storage water pressure does not produce this theoretical disadvantage.

The 1% to 2% difference (equivalent to 1.8 and 3.6 litres) between medium and high pressure cylinders with water at a temperature higher than 40^0C (>40 degr.C.) is not conclusive enough to favour one type of storage pressure over the other. The effectiveness of the internal baffle plate over the water inlet and the side entry in reducing the speed of cold water entry (thereby assisting in maintaining the integrity of the hot water layer) is as such well demonstrated.

Figure 2.17 illustrates the temperature distributions obtained after 90 and 120 litres draw-off for the medium pressure and high-pressure situations; the difference is clearly minimal.

2.9.2 Cylinder Water Temperature

From *Figure 2.18* it can be seen that volume of the piston flow zone is relatively steady at 11% (20 L) for flowrates of 4 and 8 litres per minute; this is irrespective of the initial water temperature or cylinder storage pressure. Only when the flowrate is at its highest (15 L/min) and the thermostat setting is turned down do we see a reduction in its zone size occurring.

Figure 2.27 helps to illustrate this last point; it shows the temperature distributions in the cylinder with 60, 90 and 120 litres having been drawn off, repeat plots having been made for water heated to 50^0C and to 90^0C (with the lower element). The two curves for 90L draw-off show that the piston flow zone finishes at 0.30m for the 90^0C case and at 0.25m for the 50^0C case; a difference of 5%.

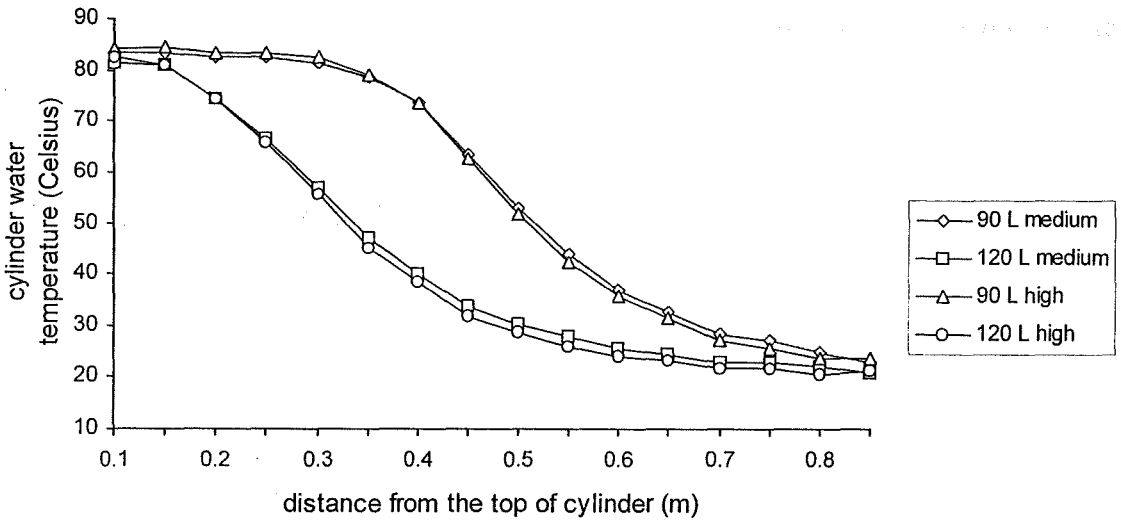


***Figure 2.17* - The difference in temperature distributions for medium and high pressure (flowrate 8 l/min, lower element heating).**

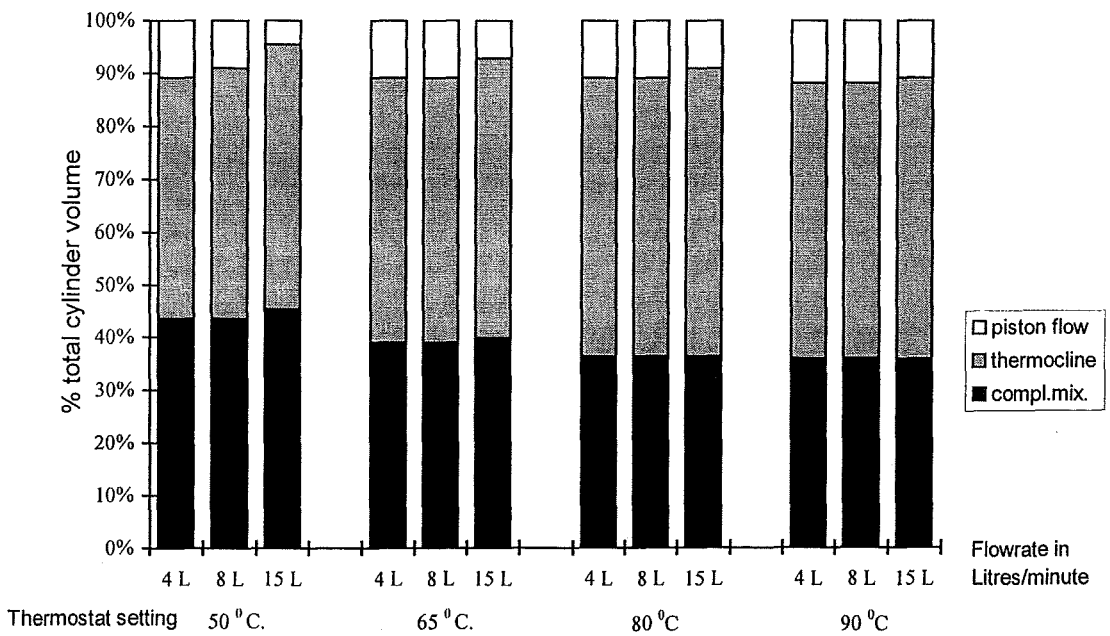


***Figure 2.18* –Piston flow, thermocline, compl.mix. volumes vs. flowrates; plotted for four different Cylinder water temperatures (medium pressure, 120 litres drawn off).**

In general increasing the thermostat temperature setting of the cylinder has the greatest effect on the thermocline volume; this quantity grows a minimum of 5% and a maximum of 8%, at 90^0C (c.f. 50^0C) depending on the draw-off rate. Correspondingly there is an equivalent reduction in the size of the complete mixing zone.

Figure 2.24 demonstrates what happens to the temperature distribution in the cylinder when the water is heated to different temperatures using the upper element. Only here we notice that the size of the piston flow volume does not vary significantly; the piston flow zone extends as far down as 0.4 metre from the top of the cylinder irrespective of the water temperature. The other two zones vary with the bottom of the thermocline going from around 0.64 m to 0.7 m as the temperature increases; a difference of 6%.

From the viewpoint of the domestic consumer it is more interesting to look what happens to the available quantity of 'useful' hot water; water with a temperature exceeding 40^0C. As expected the increasing thermostat temperature provides more of this $>40^0$C volume; from *Figure 2.19* we see that at 50^0C, with two-thirds of the initial water volume drawn off, there is on average a quarter of a tank (45 L) available. This increases to an average 40% (72 L) when the thermostat is set to 90^0C.

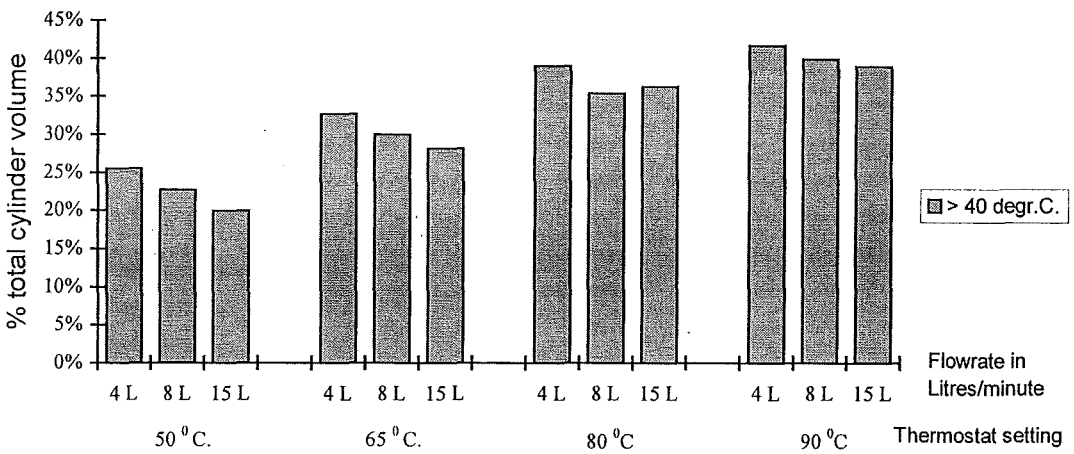


***Figure 2.19* - The volume of the > 40^0C zone vs. Flowrates; plotted for four different temperatures (medium pressure, 120 litres drawn off).**

Stated in another way; taking 50^0C as reference (not many people will be able or even want to set their cylinder thermostat lower than this), increasing the thermostat temperature by 80% to 90^0C augments the corresponding $>40^0$C volume by 60%.

2.9.3 Flowrate

The three flowrates used in the trials were 4, 8, and 15 litres per minute. The value of 8 L/min. is representative of the hot water flow for a shower; 15 L/min. is more typical for a bath.

During draw-off, when the cold water inlet jet attempts to upset the stratified layering inside the cylinder, it was expected that less internal mixing would be observed in those cycles where the water was heated to a higher degree. The theory being that water of a higher temperature possesses an equivalently lower density; and the greater the density difference between the cold inlet water and the hot stored water the greater buoyancy force on the hot water. The greater buoyancy forces suppress mixing to a greater extent; hence the volumes of the piston flow and thermocline zones would tend to remain intact.

In general this theory seemed to hold true. *Figure 2.18* shows that only for the lowest thermostat temperature of 50^0C does a reduction in piston flow and thermocline volume take

place, becoming even more pronounced when the flowrate exceeds 4 L/min. When the thermostat is set to 65^{0}C it takes the highest flowrate of 15 L/min to affect the two zones.

Only when the water temperature is boosted up to 90^{0}C (a temperature not available on the standard domestic thermostat, which is limited to 80^{0}C) is the influence of the highest flowrate negated.

Figure 2.19 illustrates the influence of flowrate on the available quantity of hot water at >40^{0}C; unlike the piston flow and thermocline volumes the effect of a faster flow-off is more obvious on the 'useful' hot water; there is an across the board reduction in its volume. Only at the higher thermostat settings do the 8 and 15 L/min flowrates produce similar results, though still lagging behind the 4 L/min. There is no doubt that a slower flowrate is beneficial in maintaining stratification.

Figure 2.20 shows again the influence of flowrate on the available >40^{0}C water (thermostat set at 65^{0}C, an average for most households); the increased turbulence of the faster flowrates has a definite reducing effect resulting in a maximum difference of 15 litres of hot water extra being present when a slower flowrate of 4 L/min is used c.f. 15 L/min.

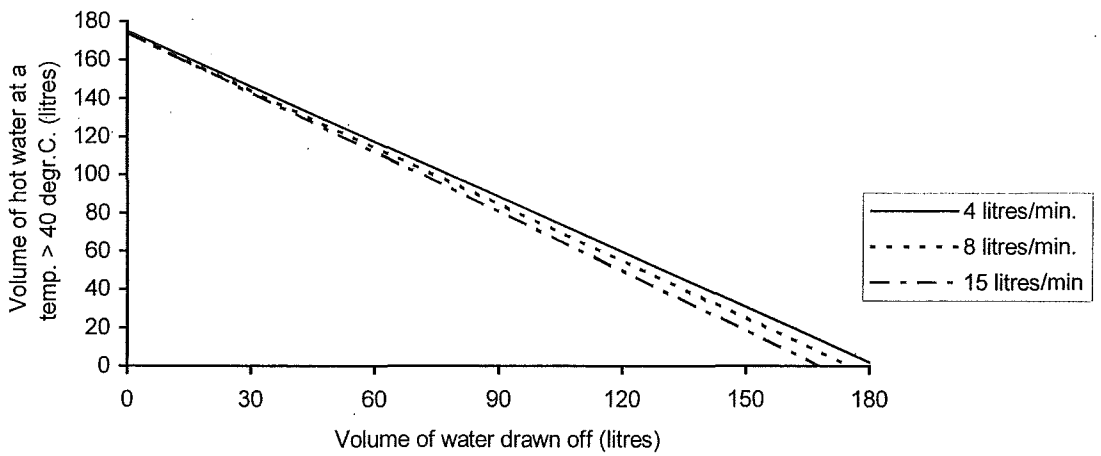


***Figure 2.20* - The volume of 'useful' hot water available with 3 different flowrates (Medium Pressure Cylinder, 65^{0}C).**

2.9.4 Re-heating (with a remnant of hot water present at the top of the cylinder)

Of interest in this trial was whether a small amount of remaining hot water, in the order of 45 litres (25% by volume), situated at the top, would disperse uniformly throughout the cylinder when the lower heating element is re-heating the cylinder water. This is a situation typically encountered late at night when lower rate electricity becomes available and cylinders start to boost their partially used hot water contents.

Figure 2.21 depicts the top and bottom water layer temperatures over the time it takes to reheat the lower region with 20^{0}C water to beyond 55^{0}C. Initially we have approximately 45 litres of water at 55^{0}C at the top of the cylinder when the clock starts to mark time and the lower heating element is switched on. As the element heats the colder water a decrease in the top layer temperature takes place. This decrease is unlikely to be caused by dispersing currents set up by the heating element, but rather is the result of a heat transfer to the colder region below. A view supported by the fact that a 7^{0}C loss occurs over a period of one hour,

which is commensurate with the 8^0C reduction in temperature in the same time period, shown in *Figure 2.23* (see section 2.9.5) below.

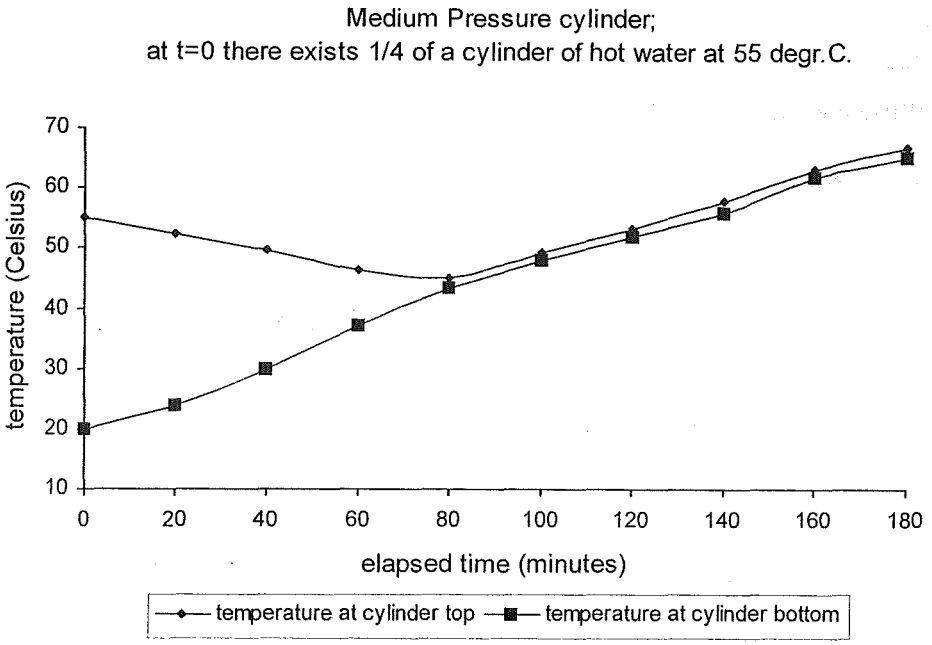


***Figure 2.21* - The influence of re-heating on the cylinder's top and bottom temperatures.**

Figure 2.26 presents the same results as for *Figure 2.21* but with additional temperature profiles shown as obtained down the length of the cylinder. Again it shows that the volume of hot water remains intact but loses 8^0C over a period of 80 minutes before starting to increase in temperature in line with the total cylinder contents. This could indicate that the upward pressure caused by the density difference of the hot and cold water is instrumental in minimising the dispersing effect of the water-heating turbulence.

2.9.5 Cylinder Standing Heat Loss

The cylinder is insulated according to New Zealand Standard 4602:1988. The trial results show that the 50mm foam insulation permits a temperature loss of 9^0C to the surroundings (which is at an average ambient temperature of 20^0C) in the first 24 hours; a result obtained with a full cylinder of hot water at an initial temperature of 70^0C. No draw-offs were made during the period of observation. *Figure 2.22* shows the typical decay curve produced by the gradual reduction in stored water temperature.

The result of a repeat of the test with half the cylinders' hot water volume drained and the remainder left standing can be seen in *Figure 2.23*, the presence of lower temperature water in the complete mixing region results in a greater rate of heat loss; within a 2 hour period the highest initial temperature has dropped 8^0C. It is interesting to observe that the rate of heat loss to the thermocline lower region and the complete mixing upper region is the greatest contributing factor in the initial reduction of the piston flow zone temperature. The heat loss to ambient air through the insulation layer of the cylinder occurs at a much slower rate and only becomes the dominant factor at a later stage.

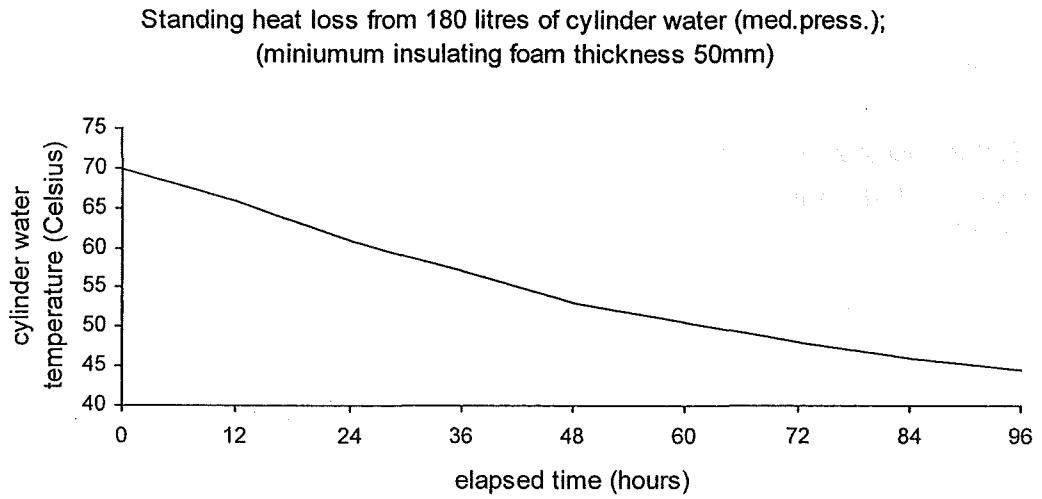


Figure 2.22 - **The reduction in the overall stored water temperature due to heat loss to the ambient environment.**

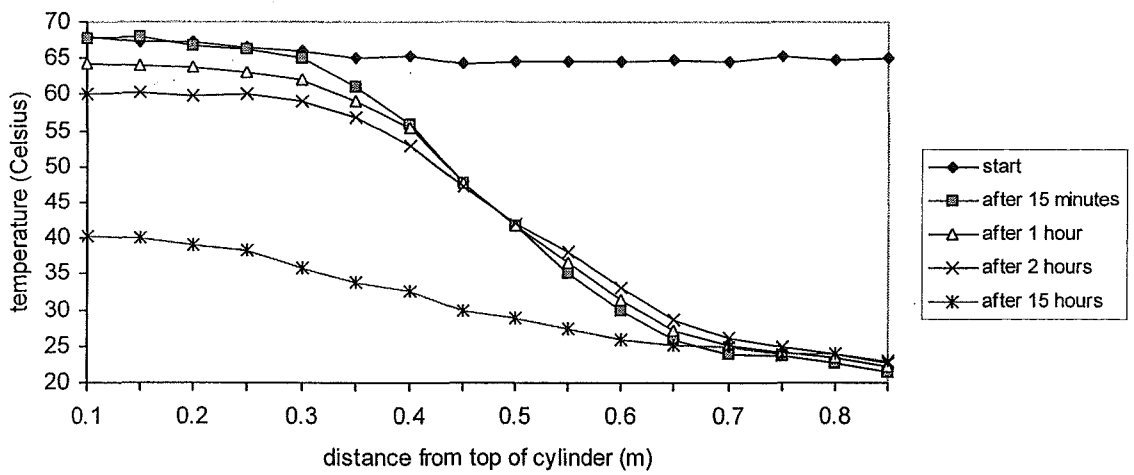


Figure 2.23 - **The cylinder's (declining) temperature distribution with time (initially heated to 70^0C; then 90 L drawn off).**

2.9.6 Intermittent versus Continuous use

Apart from 15 minute pauses to obtain a reliable cylinder temperature profile all the trials were conducted with an almost continuous draining of hot water, this is of course an unlikely domestic situation. With intermittent use the cylinder's hot water in a typical domestic situation only starts to experience a fast temperature drop if a large amount of hot water is drawn off and is replaced by cold water. Typically the period when showers are taken in New Zealand is in the morning. If heating of the cylinder contents only takes place during the night, with the lower electricity rates then available, this situation is encountered. During the daytime the heating element cannot be activated. The element's switching is remotely controlled 'ripple-control', by the distribution authorities.

Figure 2.23 is representative of the loss in temperature experienced by the contents of a partly drawn off cylinder. As the remaining volume of hot water decreases and is left standing for even longer periods the heat loss from the upper layer becomes greater yet.

If the period of major draw-offs is shifted as close as possible to the time of re-heating (e.g. late evening baths or showers) then this relatively quick drop in average water temperature can be reduced significantly.

2.9.7 Bottom element versus Top element

The location of the top element approximately 1/3 of the distance from the top of the cylinder allows for a relatively quick boost of 60 to 65 litres of water to any desired temperature. *Figure 2.24* portrays the typical temperature distribution obtained when a cylinder of cold water is heated to 2 different temperatures with the top element.

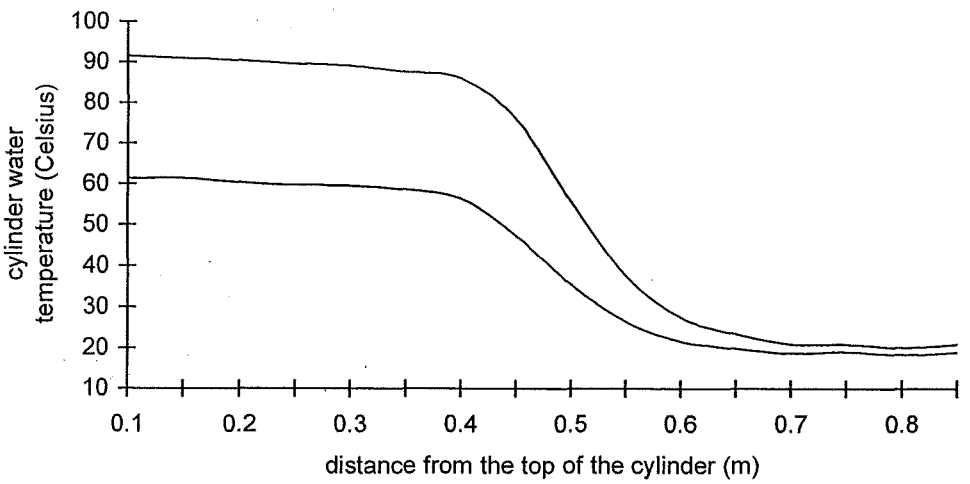


Figure 2.24 - **Typical temperature profiles established when using the cylinder's upper element to heat water to either 60°C or 90°C.**

The disadvantage here is that this volume of hot water will experience a faster heat reduction, as once again there is a large body of cold water present to encourage heat transfer. For the upper element there appeared to be no advantage in having either a high pressure or medium pressure cylinder, all the trials gave near identical results.

There is however a form of improvement with the presence of a reduced thermocline volume when compared to lower element heating. If we look at the situation where the water has been heated to 90^0C with the upper element then a piston flow volume and thermocline volume of 32% each exists prior to any draw-off. A similar situation having used the lower element for heating to 90^0C and then drawing off 120 litres of hot water, leaves a temperature distribution with a smaller piston flow zone (11%), and a much larger thermocline volume of 53% (see *Table* 2). Clearly there is approximately double as much hot water available again for the upper element case.

2.10 Conclusions

In general a faster draw-off rate causes more turbulence inside the hot water cylinder and results in small reductions in remaining hot water. This progressively worsens as a greater volume of hot water is drawn off. This detrimental process can be slowed down by storing the cylinder water at a higher temperature because, regardless of the flowrate, the trials indicate that increasing the storage temperature from 50^0C up to the highest tested 90^0C has a beneficial effect on the piston flow volume. Thus more hot water becomes available in the cylinder.

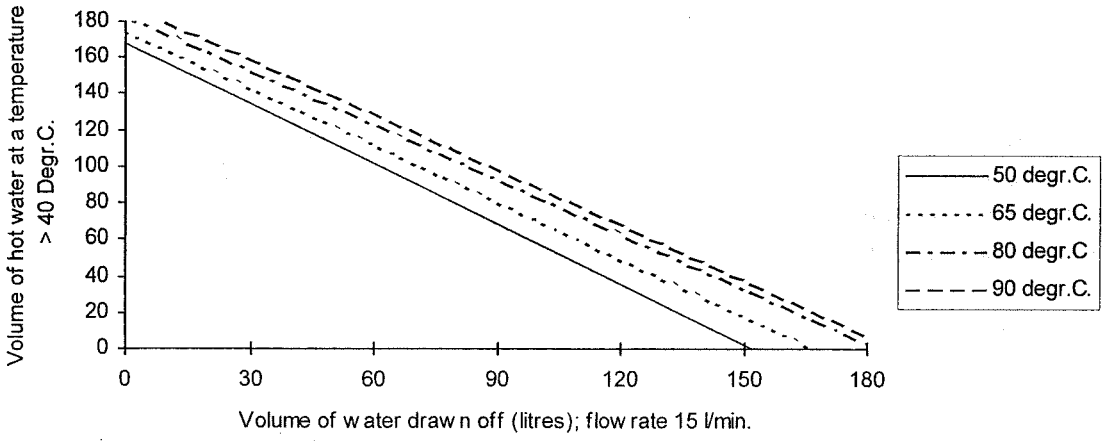


Figure 2.25 - **The volumes of 'useful' hot water available with the cylinder's thermostat set to four different temperatures.**

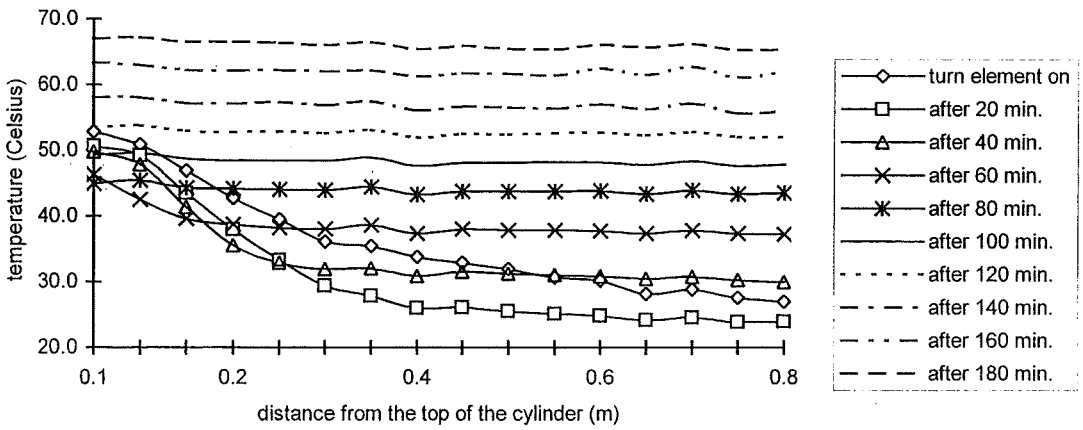


Figure 2.26 -**The temperature distributions during re-heating of a small volume of hot water at 55°C to a new temperature of 70°C.**

The present N.Z. standard 4602:1988 for insulation thickness allows a full tank of 70^0C hot water to stand unused for 24 hours with a resultant 12% drop in temperature.

It is appreciated that domestic water consumption patterns can vary a great deal (Hendtlass, 1981). Taking this into account the key issue is how the domestic consumer will get the best value for the money in heating the water in the storage cylinder:

- Large families with heavy hot water use will benefit by heating the total cylinder volume to a higher temperature and installing a tempering valve to avoid scalding.

- Single occupants or other low water users could find that utilising the upper element will suffice. For the latter it is especially beneficial to heat the water as close as possible to the time of actual utilization to avoid temperature loss in the upper layers; with the upper element it typically takes $1^1/_4$ hours to heat 65 litres of 20^0C water to 60^0C.

- High flowrates should be avoided (shower and bath); again, increasing the water temperature should help towards this as it lessens the mixing of hot and cold zones of water.

- During draw-off medium pressure cylinders have a small advantage in increased 'useful' hot water availability, over high pressure cylinders.

When analysing hot water cylinder performance for the purpose of either improving its design or determining an energy input strategy, it is important to consider the stored hot water temperature and the related actual energy content.

A single (average) temperature figure for the water stored, although technically correct, is a little misleading as the water temperature will of course vary according to its position in the cylinder. The difference in regional temperatures is most significant immediately after draw-off has occurred. Although an average water temperature is indicative of what remains in the cylinder, it fails to convey an accurate picture; is there enough hot water for a shower for instance? For a control system a single energy figure is perceived as a better means of representing what the cylinder is still capable of delivering. It is also a more conventional means of expressing efficiency and possible losses.

Therefore an improved control strategy is not to look at average temperature as means of control but to concentrate on the energy content, closely related to the timing of the applied energy.

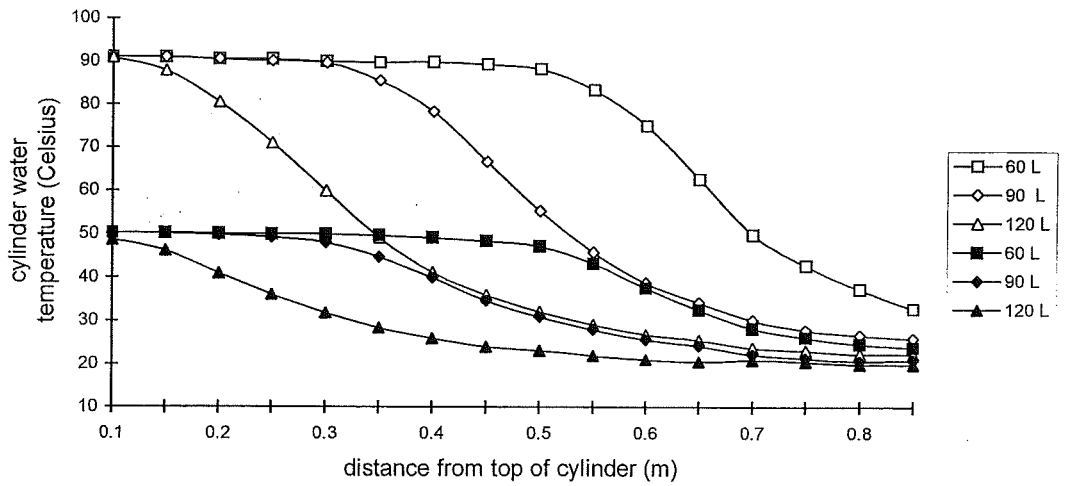


Figure 2.27 - **Temperature distributions for 50 and 90°C thermostat settings showing 60, 90 and 120 L draw-offs (flowrate 15 L/min).**

2.10.1 Implications for an Energy Management System

The energy management system will focus on controlling the thermal energy content of the cylinder water by increasing or decreasing its temperature to suit the energy demand of the domestic consumer. The thermal energy content of past daily used hot water quantities will determine the overall temperature of the next day's predicted amount of water.

Consideration could be given to adjusting the storage pressure in order to be able to raise the temperature, and thus energy content, to a higher set-point. This should also allow smaller hot water cylinders to cope with larger consumer demands, a problem that in the domestic situation is normally solved by installing an increased size cylinder.

The results obtained clearly indicate that control over the hot water in the cylinder lies in manipulation of the water *temperature* and not the *quantity*. That is unless some form of heating element can be installed that allows adjustment of its position along the entire axis of the cylinder (with the resulting increase in complexity and costs), or unless an effective means can be found of inserting the desired quantity of hot water at the correct temperature on top of an undisturbed cold water layer. The aim of the energy management system is to reduce the amount of electricity used by a domestic household by only heating up that

amount of water that such a household is likely to need in any given 24-hour period. A possible method of being able to provide the requisite daily amount of hot water is by monitoring and modifying the *thermal energy* content of the water in the cylinder. To heat only the required level of hot water for the consumer, without an excessive surplus or shortage, it is essential to find a means of predicting how much thermal energy a domestic consumer will use in the next 24 hours, based on a learning curve of the use of hot water from a cylinder over a period of weeks or months. The forecasting method chosen will be expressed in a software algorithm. Additionally the software will also need to be able to control the thermal energy added to the water and therefore needs to have an idea of the time and water temperature.

As such, part of the software is a further development of the acquisition program previously written for the express purpose of testing the hot water cylinder. It is further envisioned that this component of the software not only takes on the role of the customary thermostat[3], but also provides a number of automated features. Some suggestions are: an infinitely variable temperature setting, minimum hot water detection, Legionella bacteria inhibiting routines, equipment failure detection, reduced standing heat losses by heating time optimisation, emergency boost operation and spot-price tariff activation. As such it will make good use of the behavioural data obtained previously.

Chapter 3 then, focuses on what is essentially a more complex time-series estimation problem; a means of predicting the amount of hot water likely to be used by a domestic household in the near-future time period.

2.11 Summary

Using a readily available domestic hot water cylinder as a basis, the essential data acquisition and control equipment have been presented. This equipment has both been designed and assembled or else purchased where this was justified. The next step was interfacing the different equipment in order to build an automated and fully instrumented test rig. Finally a dedicated software program was shown developed, which carried out the data acquisition and controlled the cylinder and associated peripheral equipment.

Subsequently the results of the data acquisition were expounded upon and illustrated by means of a series of tables and figures.

[3] It also alleviates the need for 'ripple control' actuated relays and possibly even the ripple signal itself, as the internal clock of the system can determine the Distribution Authority's periods of low cost electricity.

Chapter 3. Linear prediction of a discrete time series

3.1 Introduction

In the previous chapter the research concentrated on determining the behaviour of the water content in a 180 litre domestic cylinder and on the design and appraisal of the data acquisition and control equipment. In the conclusions it was noted that by selecting one of two available methods it is feasible to govern the amount of *stored energy* (in the form of hot water) available from the domestic cylinder.

The available methods were 1) heating a (variable) *portion* of the available volume of water in the cylinder or 2) heating the *total* volume of water in the cylinder to a varying *temperature*. The justification for using energy as a measure of daily water usage is that it is convenient. It not only takes into account the amount of water used but also the decrease in temperature due to losses to the direct ambient environment and to the "dead legs".

Altering the heated volume of water is at first glance the most straightforward; but in actual fact offers little scope as the standard domestic cylinder has a fixed position element and will therefore always heat approximately the same volume of water. The second method, regulating the temperature, is what is partially being accomplished in existing cylinders by the thermostat; the problem here being that the thermostat is usually set at a desired value when first installed (anywhere between 55 and 85 ^{0}C) and is seldom altered after that. However, it is the most practical method without significantly altering the existing (proven!) cylinder design. What it really needs in order to be successful is a better means for both monitoring and, most important, altering the temperature of the stored water. This then, would constitute a viable means of controlling the available energy output of a cylinder.

With the existing test equipment (see *Chapter* 2) and appropriate control software it should be possible to dynamically vary the amount of hot water energy stored in the cylinder. And the *amount* of energy that needs to be stored could, for instance, be determined by forecasting the likely amount of hot water to be used in, say, the *next* 24-hour period. Such a forecast could conceivably result in a decrease, but also an increase, of stored energy for a conventional cylinder (whose thermostat is more than likely set at a arbitrary value without taking into regard the actual water use). The situation where the stored hot water energy is increased will occur if there is a tendency for the consumer to run out of hot water on a regular basis. Here the assumption is made that the consumer would rather have the system meet the demand, despite a possible increase in electricity costs, than risk running out of hot water. (The possibility of giving the consumer a choice in this respect is further explored in *Chapter* 9 with the 'eFEMS').

Combining this energy/temperature control with a *time series* based method of prediction is the next step in developing an energy management system that is able to manipulate the heating of water or, equally important, other fluid mediums. It seems logical that in predicting *future* demand the *historic* usage values as related to a 'standard' domestic household would need to be examined, in order to arrive at some conclusion with regards to typical demand patterns.

Subsequent research found that insufficient historical data was available to be able to test any forecast model in a rigorous manner. *Figure 3.1* shows generalised consumption values produced by Hendtlass in his report for the University of Canterbury in 1981. It was thus deemed worthwhile to collate real hot water demand data values over a significant time span and verify the suitability of the chosen model accordingly.

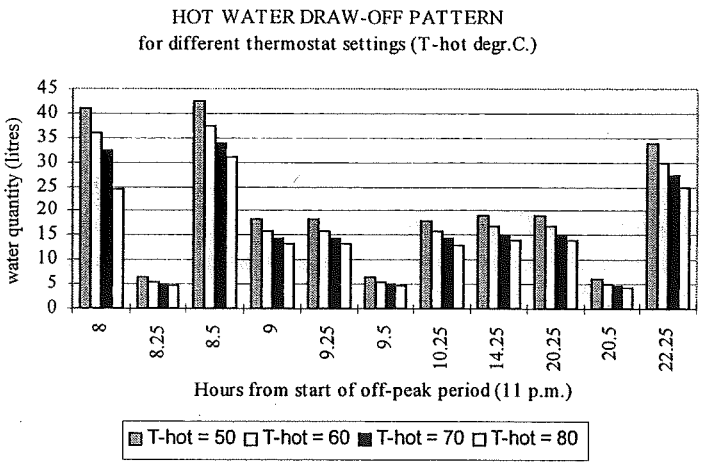


Figure 3.1 – **Typical hot water demand profile for a family of four persons.**

An unknown factor, but of great importance in selecting a prediction model, was whether the any demand data series would have a *linear* or *non-linear* character. As a starting point it was assumed that any time series formed by the observed water use was of a *linear* nature, meaning that the next value of the time series could be determined by a linear combination of the previous values. This allowed the potential utilisation of a large number of proven forecasting methods. Should the initial assumption prove incorrect than a *non-linear* approach to the problem could be investigated as a next step.

3.2 The prediction model

Prediction, or forecasting, is a process in which one is concerned with a sequence of data points, a *discrete time series*. Some or all of the data in this sequence have been observed and are known. The prediction problem is to build a model, usually in the form of a mathematical function, from which future data in the sequence can be drawn. The required model should give a good representation of the data in any local segment of time. The model need not represent the data a very long time in the future. In buying petrol for the car, one is not concerned with the fact that in five years time the car's petrol consumption might have dropped from 15 km/litre to perhaps only 10 km/litre. The phenomenon of decreasing consumption will be taken into account by revising the prediction in the light of later experience. This is analogous to predicting the hot water energy demand in a household; a decrease in the number of occupants at *a much later date* has no bearing on the demand in the immediate future.

The discrete time series represented by the observed daily hot water use should be primarily stationary when viewed in the short-term period. This assumes that data of a number of months ago is not relevant to current conditions (which is not entirely correct if there are, for instance, seasonal influences; this is discussed in *Section* 3.3). As the prediction is to be revised with each new observation, it should adapt to changes in water demand over a period of time. This process can be visualised as a graph of the data with a window superimposed on the later values (*Figure 3.2*). As time passes the window will move to the right, the future, so that the most current observations are visible; points in the past will have disappeared from view.

It may be more of challenge to identify a model that represents the data over a longer span of time, as the time series could then exhibit more complex non-stationary characteristics. If the

model is a poor representation then the distribution of forecast errors will have a large variance. A certain amount of improvement can be obtained by more elaborate models and more refined forecasting techniques (Brown, 1963).

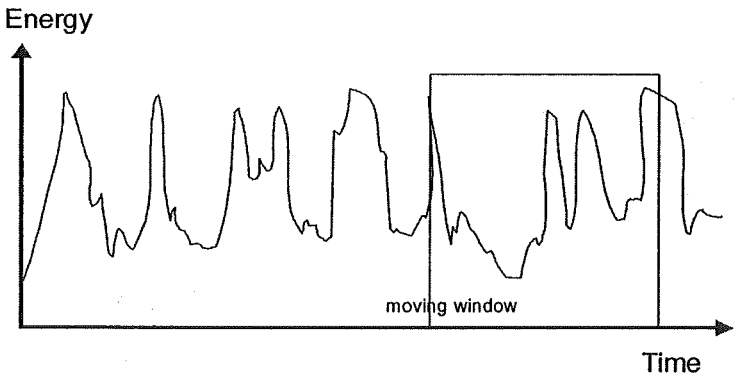


Figure 3.2 - **Illustrating the concept of a moving window.**

3.3 Predicting time series with models - a statistical viewpoint

A time series typically consists of a set of observations on a variable y_t, taken at equally spaced intervals. As an example consider *Figure 3.3*, where y_t, the logarithm of the national electricity consumption is plotted against weeks t = 1, 2, ..., 730, where week number 1 corresponds to the week starting from January 4th 1970 (Walkington, 1990).

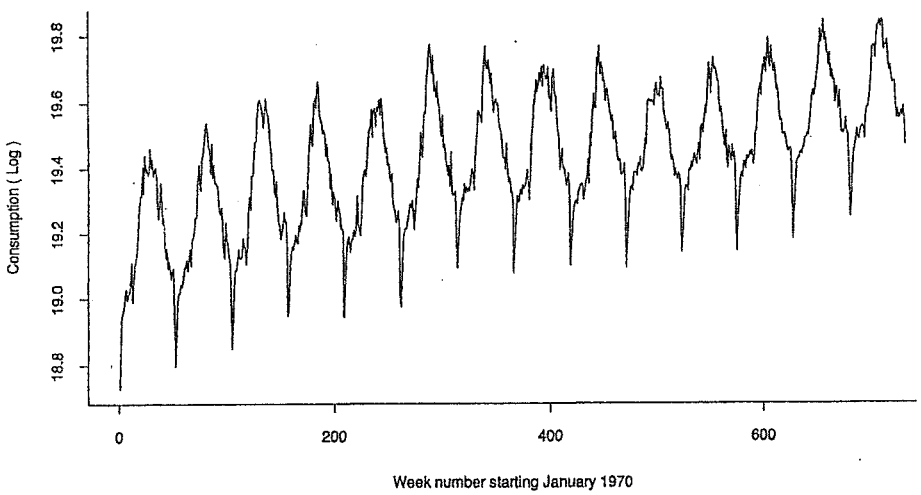


Figure 3.3– **National weekly electricity consumption 1970 - 83**

The question of whether it is possible to produce accurate forecasts from a time series of this nature has led to the development of a considerable number of different models. These models attempt to explain the movement in y_t in terms of its past values or by its position in relation to time. Predictions are then made by extrapolation.

The literature on model- or system- identification is extensive; a good practical introduction is given by Ljung et al. (1994). Model identification has its roots in standard statistical techniques and many of the basic routines have direct interpretations as well known statistical methods such as least squares and maximum likelihood. The control community for one took an active part in developing and applying these basic techniques to dynamic systems right after the birth of "modern control theory" in the early 1960's. Maximum

likelihood estimation was applied to difference equations (ARMAX, auto-regressive moving average with extra inputs, -models) by Åström et al. (1965), and thereafter a wide range of estimation techniques and model parameterisation flourished.

Notable among these models are those for stationary processes including *autoregressive* and *moving average* models; models based on the decomposition of a time series into a trend, seasonal and irregular components, and the *exponentially weighted moving average* procedures of Holt (1957) and Winters (1960). Other forecasting methods considered have included *regression* forecasting, where use is made of the relationship between the variable to be predicted and other variables that may explain its variation, and *spectral* models where an analysis is made of the tendency for oscillations of a given frequency to appear in the data.

Visual inspection of graphs of time series often reveals *trends* and *seasonal variations*. These are important features of the data and it seems desirable to model these features explicitly. This is not to suggest that such a model is necessarily representing the manner in which the time series evolved but rather that it models these observable features.

Such structural models lie behind a number of seasonal adjustment methods where the aim is to remove the seasonal component from an observed time series. Techniques used in such methods as X-11 as developed at the U.S. Census Bureau (Shiskin et. al., 1967) generally involve the use of moving averages (or linear smoothing filters) to smooth the seasonal fluctuations from the data, leaving just the trend and irregular components.

An alternative approach is to specifically model the various components (Walkington, 1990). The simplest models are just deterministic functions of time; for example, a linear trend model and a seasonal indicator model where the sum of the seasonal parameters are constrained to be zero. Thus in the case where there are S seasons

$$y_t = \alpha + \beta t + \sum_{j=1}^{S} \gamma_j z_{j,t} + \varepsilon_t \qquad t = 1, 2, \ldots, T \tag{3.1}$$

Here the trend is modelled as being linear with level α and slope β. The seasonal component $\Sigma \gamma_j z_{j,t}$ repeats every S time periods. The $z_{j,t}$'s are seasonal indicator variables which equal 1 if time t is in season j or 0 otherwise. The irregular component is generally modelled as being a sequence of uncorrelated, normally distributed random variables with mean zero and variance σ^2 (i.e. a normally distributed white noise process).

This model can be written in the general linear form $y = X \beta + \varepsilon$ and then estimation is possible through *least squares regression* techniques. The estimated model can then be used to provide forecasts for future values of y_t. However models such as this, that include *globally* constant parameters, have been found to be of little value in forecasting applications. The difficulty with such global models for forecasting purposes is that they are assumed to hold at all points in time with the parameters remaining constant throughout. In applications it has proved very difficult to fit such models; for example, consider the electricity consumption data from *Figure 3.3* and attempt to fit a straight line through the middle of the entire series. It is clear that an allowance needs to be made in such models for the components to evolve over time.

This weakness displayed by globally constant models led to the development of *locally* constant models. The idea that the more recent observations should be given more weight when forecasting led to the development of the *exponentially weighted moving average*, first given by Holt (1957), where the prediction of the next observation is as the prediction of the previous observation plus a proportion of the prediction error of the previous observation.

In the years between the mid-sixties and the early eighties, the focus of attention for models of *non-stationary* time series had not been on such structural models but rather more on such approaches as *auto-regressive-integrated-moving-average* (ARIMA) modelling. Here the time series were transformed, usually by differencing, with a view to reducing them to stationary processes where they could be modelled as auto-regressive-moving-average (ARMA) processes. Box and Jenkins (1976) developed an approach to time series forecasting based on ARIMA models.

The *state space* model was first outlined by Kalman (1960) in the engineering literature. Through what has become well known as the *Kalman filter*, the equations allow optimal predictions of future observations to be made. The statistical aspects required in the modelling, including estimation, prediction and smoothing can be handled with relative ease. Harrison and Stevens (1976) in their *Bayesian* forecasting procedure make use of the Kalman filter and the 'Dynamic Linear Model' (i.e. state space form). They view the Kalman filter results from a Bayesian viewpoint as a way of updating a *prior* distribution for the state vector to get a *posterior* distribution.

Harvey in his book "Time Series Models" (1981) devoted an entire chapter to state space models. He aimed " to make the material, previously confined almost entirely to the engineering literature, more accessible to those with a background in statistics or econometrics".

The state space model considered is

$$y_t = \mathbf{z}_t^T \alpha_t + \varepsilon_t \qquad t = 1, 2, \ldots, T \qquad (3.2)$$

$$\alpha_t = \mathbf{T}\, \alpha_{t-1} + \eta_t \qquad t = 1, 2, \ldots, T \qquad (3.3)$$

where the observed variable y_t is related to the state vector α_t by the measurement (3.2) and the state vector, although not directly observable, is assumed to be governed by the transition equation (3.3). The $\mathbf{z}_t$ vector and $\mathbf{T}$ matrix are assumed fixed. The ε_t are white noise disturbances and η_t is a vector of disturbances with mean zero. All the disturbances are taken to be serially uncorrelated and furthermore, they are uncorrelated with each other for all time periods. It is the elements of state vector α_t that are of interest. In fitting the model the task then is to use all the data available at time t (i.e. $y_1, y_2, \ldots, y_t$) to find the best estimate of α_t, the state vector at time t.

The Kalman filter equations have been shown to provide such optimal estimates of α_t. Applied in a recursive manner as each new data value arrives, the filter calculates the estimate of α_t using all the data available at time t and then α_{t+1} using the data up to time t. The one step ahead forecast of y_{t+1} follows from equation (3.2).

Ljung (1996) describes this as the *subspace projection* approach to estimating the matrices of the state space model, including the basis for the representation and the noise covariance matrix. Overschee et al. (1994) and Larimore (1983) covers a number of variants on this approach suited to multi-variable systems.

Another important model for the purpose of forecasting is that of *spectral analysis*. When the process to be predicted is *periodic* it is possible to describe it in terms of sines and cosines. By utilising the *Fourier series* a reasonably periodic function of time could well be represented by taking a sufficient number of terms in the series. The intention here is to limit the number of terms to only those that are necessary and really significant (Schoukens et al., 1991). When the values of the coefficients in the model have to be estimated, at least part of the computing effort rises as the cube of the number of degrees of freedom. Equation (3.4) shows a time series y(t) with zero mean, several periodic functions plus some additive noise.

$$y_t = a_1 \cos \omega_1 t + a_2 \cos \omega_2 t + \ldots\ldots a_n \cos \omega_n t + \varepsilon_t \quad (3.4)$$

The frequencies ω_i are all distinct from each other. A more general formulation would include sine terms as well, to allow for relative phase shifts of the various frequencies. The *power spectrum* is utilised in detecting the frequencies ω_i that should be included in a periodic model. The power spectrum is a cosine transformation of the *autocovariance* function, which incidentally can be used to design the optimum linear forecast filter for an autoregressive process (Brown, 1963). The autocovariance is the average (over all time) of the lagged products

$$R_{yy}(k) = \lim_{T\to\infty} \frac{1}{2T+1} \sum_{t=-T}^{T} y_{t+k}\, y_t \quad (3.5)$$

The power spectrum will be a continuous function of frequencies $P(\omega)$. It is defined as the Fourier transform of the autocovariance function.

$$P(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} R_{yy}(t) \cos \omega t \, dt \quad (3.6)$$

It can be shown that the autocovariance function has *local maxima* at the periods of the various frequencies present in the time series. At those points, the value of the covariance function will be

$$R_{yy}(k_j) = \frac{1}{2} \sum_{j=1}^{n} a_i^2 \cos \frac{2\pi\omega_j}{\omega_i} + R_{\varepsilon\varepsilon}\left(\frac{2\pi}{\omega_i}\right) \quad (3.7)$$

A plot of the autocovariance function might show the maxima, but it will be hard to tell the contribution from one particular frequency. Taking the Fourier transform, the power spectrum will be

$$P(\omega_i) = \frac{a_i^2}{2} \quad (3.8)$$

With this function it is much easier to determine the frequencies present and their amplitudes. It is important to know all frequencies with significant power, so that the appropriate terms can be included in the model.

The practical application of this technique can be shown using a discrete data series as represented by *Figure 3.4*. The plot of the data series immediately shows some significant features. The first one is that it is a rising pattern, on the average, that might be described by a linear function of time. Superimposed on this basic trend is very definite seasonal cycle. In addition it is clear that the amplitude of this seasonal variation is also growing. The seasonal pattern is not exactly a sine wave and would indicate the presence of higher harmonics in the waveform.

The autocorrelation coefficient for *zero* lag would equal 1, the largest value the coefficient may attain. The smallest possible value is –1. Generally, for noisy data, the autocorrelation coefficients would drop down to zero for any substantial lag. Pure random noise would have zero correlation between samples not identically equal to each other.

Figure 3.5 shows two autocorrelation functions computed from the data. There is a significant trend, which shows as a high correlation for all lags in the raw data. When the trend is removed by subtracting the current average rate the periodic nature of the data is much clearer.

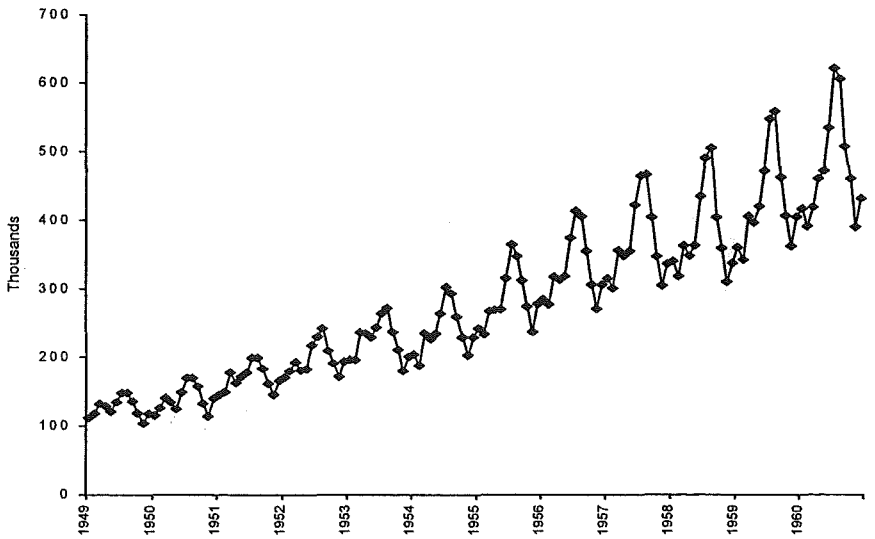


Figure 3.4 – **U.S. international airline passengers**

The square bottom of the deviation from the mean autocorrelation function is an indication of higher harmonics in the waveform. Contrast this with *Figure 3.6*, however, which shows the power spectrum of the raw data. There are clear spikes for frequencies with periods 6, 12, and 21 months. Spikes of 6 and 12 months are apparent in the deviation from the mean. From this it is much clearer (i) what frequencies are indeed present and (ii) what their amplitudes are. In general it is not crucial what the power is, since the model used in prediction will estimate it. It is, however, crucial to determine all the frequencies with significant power, so that the appropriate terms can be included in the forecast model.

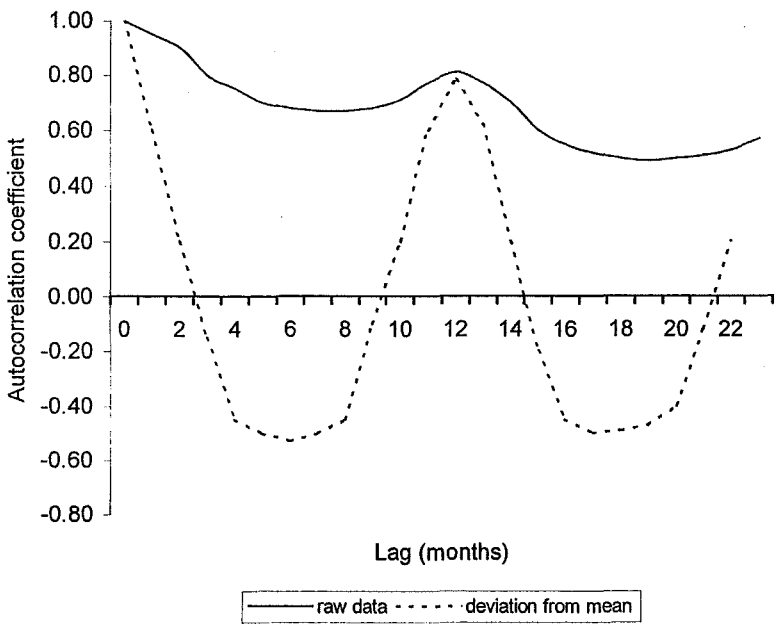


Figure 3.5 – **Autocorrelation functions for airline passenger data**

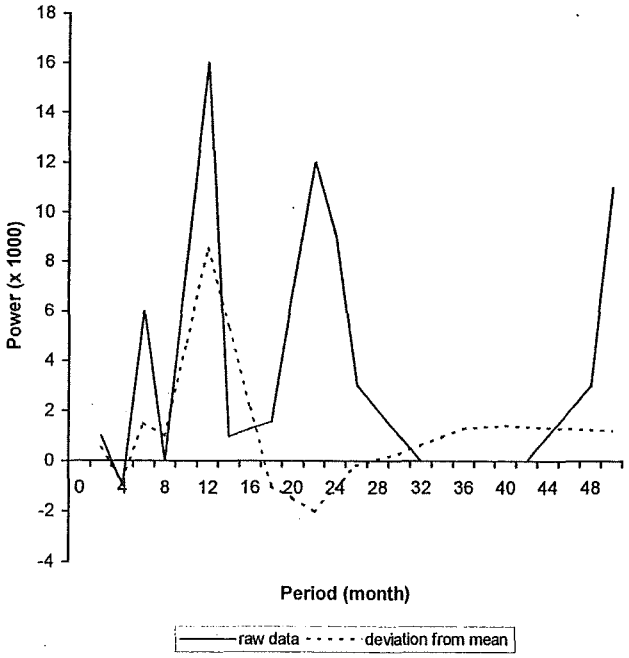


Figure 3.6 – **Power spectrum for airline passenger data**

3.4 Modelling hot water demand

A considerable number of the approaches mentioned in the previous section need to be able to draw on a large amount of historic data, forming the time series, in order to establish a reliable approximation to the underlying model. Given the fact that such information is not readily available when an energy management system is first installed in a domestic or industrial situation, it is prudent to choose a model that will be up and running with a minimum amount of historic data. The model should also be able to adapt to changing conditions given little prior information, and be straightforward to program in a universal software language such as ANSI C without having to resort to complex functions. This is desirable as an actual commercial system would have to embody some form of (inexpensive) microprocessor. The appropriate compiler for the microprocessor will limit the number of different C functions it can handle.

Previous departmental research in Energy Management by Bowling (1992) has seen the use of a simple algorithm based on the principle of *moving averages* which attempted to forecast the anticipated hot water demand for the coming period. In a simulation the program simply heated up a mean total quantity of water as used in previous 24 hour periods and ensured that this amount was ready by 7 O'clock in the morning. Each (simulated) daily used water quantity was stored in a look-up table and as more historical data became available the mean was calculated using an averaging mechanism which took into account the old data, the number of times this data had been updated, and new information from external sensory inputs.

Although it operates with little initial data and has a straightforward algorithm the used method has a considerable smoothing action which has two major disadvantages. The first one being that it fails to take note of the days when *maximum* or *minimum* demand takes place, and secondly that it is slow to react to any significant increase or decrease in hot water use; both of these changes tend to get averaged out. (A further important disadvantage in the

system was that it was based on a non-existent "fast acting" hot water cylinder[4], one which manages to provide variable amounts of *instantly* heated hot water without degrading the thermal stratification within the cylinder – see also *Appendix* B).

A spectral expansion technique, utilising the Fourier Transform, was considered to be a more accurate method for predicting hot water demand (Champeney, 1978). When considering the varying signal depicted by a plot of the quantity of drawn-off hot water against time, as in *Figure 3.7*, the data can be seen to represent a time series signal with a irregular periodicity. It is also a methodology that is well understood and has found many well-documented applications (Elliott et al., 1982)

Spectral techniques aim to characterise a known signal in terms of the frequencies present and their amplitudes, and are capable of modelling the peaks and troughs of the time series (analogous to sharp spectral features) while still providing some measure of desired smoothing (see *Figure 3.6*). They also represent an efficient computational tool allowing the data to be extrapolated and mathematically manipulated in order to incorporate a 'moving window' arrangement for reading the latest data.

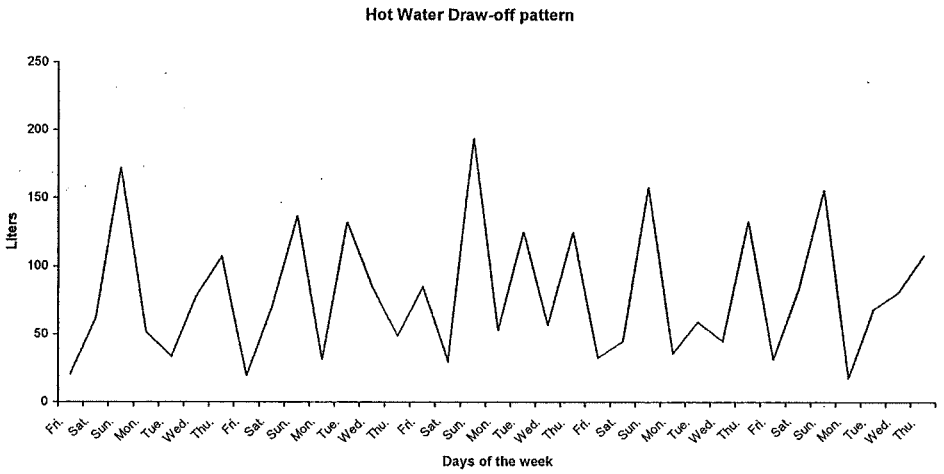


***Figure 3.7* - The quantity of hot water used (typical household) every 24 hours, over a 5-week period.**

However, the (Fast) Fourier Transform is not the only way to estimate the power spectrum of a discrete time series (Press et al., 1992). In the spectral domain, one limitation of the FFT is that it always represents a time series function's Fourier transform as a polynomial in $z = \exp(2\pi i f \Delta)$ where Δ is the *sampling interval*. Sometimes, time series have spectra whose shapes are not well represented by this form. An alternative form, which allows the spectrum to have *poles* in the z - plane, is used in the *Maximum Entropy* spectral estimation.

3.5 The Maximum Entropy Method

If the real frequency range is not limited to the Nyquist interval $-f_c < f < -f_c$, where f_c is the Nyquist Critical frequency, but also includes the entire complex frequency plane then this complex f - plane can be transformed to a new plane called the z - transform or z- plane, by the relation

$$z \equiv e^{2\pi i f \Delta} \tag{3.9}$$

[4] In fact, domestic cylinders of this type do exist although they are by no means in common use; presumably as they are more complicated, and therefore more expensive, to manufacture.

where Δ represents the sampling interval in the time domain.

The z – transform plays much the same role in the analysis of discrete time systems as the Laplace transform does with continuous time systems. Important sequences and their z – transforms are covered, amongst others, by Jury (1964).

The discrete Fourier transform of an N-point sampled function y(t), sampled at equal intervals, is

$$Y_k = \sum_{j=0}^{N-1} y_j \, e^{2\pi ijk/N} \qquad k = 0, 1, \ldots, N-1 \tag{3.10}$$

then the periodogram estimate (an estimate of the power spectrum of a function y(t) by taking the modulus-squared of the discrete Fourier transform of some finite, sampled stretch) of the power spectrum is defined at N/2 + 1 frequencies as

$$P(f_k) = \frac{1}{N^2}\left[|Y_k|^2 + |Y_{N-k}|^2\right] \qquad k = 1, 2, \ldots, \left(\frac{N}{2} - 1\right) \tag{3.11}$$

where f_k is defined only for the zero and positive frequencies

$$f_k \equiv \frac{k}{N\Delta} = 2 f_c \frac{k}{N} \qquad k = 1, 2, \ldots, \frac{N}{2} \tag{3.12}$$

Comparing (13.9) to equation (13.10) and (13.12) it can be seen that the FFT power spectrum estimate (3.11) for any real sampled time series function $y_k \equiv y(t_k)$ can be written, except for normalisation convention, as

$$P(f) = \left| \sum_{k=-N/2}^{N/2-1} y_k z^k \right|^2 \tag{3.13}$$

Equation (3.13) is not the true power spectrum of the function y(t), but only an estimate. There are two reasons for this. First, in the time domain, the estimate is based on only a *finite* range of the function y(t) which may in actual fact have continued from t = -∞ to ∞. Second, in the z - plane of equation (3.13), the finite Laurent series offers, in general, only an *approximation* to a general analytic function of z (Press et al., 1992).

A formal expression for representing 'true' power spectra (up to normalisation) is

$$P(f) = \left| \sum_{k=-\infty}^{\infty} y_k z^k \right|^2 \tag{3.14}$$

This is an infinite Laurent series that depends on an infinite number of values y_k. Equation (13.3) is only one form of approximation to the analytical function of z represented by (3.14); the form that is implicit in the use of FFT's to estimate power spectra by periodogram methods (Oppenheim et al., 1989).

Equation (3.14) can also be approximated more generally with a rational function, one with a series of type (3.13) whose free parameters all lie in the denominator, namely

$$P(f) \approx \frac{1}{\left| \sum_{k=-M/2}^{M/2} b_k z^k \right|^2} = \frac{a_0}{\left| 1 + \sum_{k=1}^{M} a_k z^k \right|^2} \tag{3.15}$$

The second equality brings in a new set of coefficients a_k's whose values can be determined by autocorrelative means. Approximations (3.13) and (3.15) are very different in character; equation (3.15) can have poles, corresponding to infinite power spectral density on the unit circle in the complex z-plane.

The Nyquist interval $-f_c < f < -f_c$ on the real axis of the f-plane maps one-to-one onto the unit z-circle; this means that the poles in (3.15) correspond to real frequencies in the Nyquist interval. Such poles can provide an accurate representation of underlying power spectra which have sharp discrete 'lines' or delta functions. By contrast equation (3.13) can have only zeros, not poles, at real frequencies in the Nyquist interval, and will thus attempt to fit sharp spectral features with, essentially, a polynomial. The approximation (3.15) bears the name Maximum Entropy Method, or MEM for short.

To obtain spectral estimates from (3.15) it is necessary to determine coefficients a_0 and a_k's from a historic data set. The autocorrelation of the sampled function y_k at lag j is

$$\Phi_j \equiv \langle y_i y_{i+j} \rangle \qquad j = \ldots, -3, -2, -1, 0, 1, 2, 3, \ldots \tag{3.16}$$

where the angle brackets denote averaging over i. With a finite set of data y_0 to y_n, the estimate of (3.16) is

$$\Phi_j = \Phi_{-j} \approx \frac{1}{N+1-j} \sum_{i=0}^{N-j} y_i y_{i+j} \qquad j = 0, 1, 2, \ldots, N \tag{3.17}$$

Thus, from N + 1 data points the autocorrelation at N + 1 different lags j can be estimated.

The *Wiener-Khinchin* theorem states that the Fourier transform of the autocorrelation is equal to the power spectrum. In z - transform language, this Fourier transform is a *Laurent series* in z. The equation that is to be satisfied by the coefficients in equation (3.15) is thus

$$\frac{a_0}{\left|1 + \sum_{k=1}^{M} a_k z^k\right|^2} \approx \sum_{j=-M}^{M} \Phi_j z^j \tag{3.18}$$

The approximate equal in equation (3.18) is meant to indicate that the series expansion of the left-hand side is supposed to agree with the right-hand side term by term from z^{-M} to z^M. Outside this range of terms, the right-hand side is zero, while the left-hand side will still have non-zero terms.

Note that M, the number of coefficients in the approximation of the left-hand side of equation (3.18), can be any integer up to N, the total number of autocorrelations available. M is called the *order* or *number of poles* of the approximation.

The series expansion of the left-hand side of equation (3.18) defines a form of extrapolation of the autocorrelation functions to lags larger than M, and even to lags larger than N, i.e. extrapolated beyond the actual data.

Equation (3.18) needs to be solved for the coefficients on the left-hand side, in terms of known autocorrelations on the right. There is a linear set of relations between the autocorrelations and the coefficients a_0 and a_k. In fact these coefficients satisfy the matrix equation;

$$\begin{bmatrix} \Phi_0 & \Phi_1 & \Phi_2 & \dots & \Phi_M \\ \Phi_1 & \Phi_0 & \Phi_1 & \dots & \Phi_{M-1} \\ \Phi_2 & \Phi_1 & \Phi_0 & \dots & \Phi_{M-2} \\ \dots & \dots & \dots & \dots & \dots \\ \Phi_M & \Phi_{M-1} & \dots & \dots & \Phi_0 \end{bmatrix} \bullet \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \dots \\ a_M \end{bmatrix} = \begin{bmatrix} a_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.19}$$

This is a *symmetric Toeplitz matrix* i.e. one whose elements are constant along the diagonals. The symmetry of the set is exploited in the algorithm and its application method can be attributed to J.P.Burg (Childers, 1978). For the software program it involves a recursive procedure for increasing the order M (the number of poles) by one unit at a time, at each stage re-estimating the coefficients a_j, j = 1, .., M.

Burg's method of solving for the coefficients is the key to utilising *linear prediction* for a time series. Linear prediction (LP), as explained in *section* 3.6, and MEM are analogous methods; MEM *characterises* a known time series signal in terms of a finite number of poles that best represent it spectrum in the complex z-plane. LP *extrapolates* the signal using its characterisation in terms of these same poles.

3.6 Linear Prediction

Classical linear prediction specialises to the case where the data points are equally spaced along a line, y_i, i = 1, 2, …, N. The general equation for predicting the next value y_n of a time series from the previous M consecutive values of y_i is given by,

$$y_n = \sum_{j=1}^{N} d_j y_{n-j} + e_n \tag{3.20}$$

Equation (3.20) states that it is possible to construct an estimate of y_n as a linear combination of the known, noisy, values. Should y_n be an *existing* point than the problem becomes one of optimal *filtering* or estimation (to remove the noise). On the other hand, if y_n is to be a *completely new* point than the problem becomes one of linear prediction. In this last interpretation, e_n is the *discrepancy* of the prediction at timestep n, i.e. the amount which must be added to the predicted value to give the true value y_n. The objective is to determine values for the *linear prediction coefficients* d_j that will optimise the relation $|e_n| << |y_n|$ for all n.

The actual N data points can be used to estimate the autocorrelation components Φ_j,

$$\Phi_j \equiv \langle y_i y_{i+j} \rangle \approx \frac{1}{N-j} \sum_{i=1}^{N-j} y_i y_{i+j} \tag{3.21}$$

Once the autocorrelation components have been calculated the following equation will allow the linear prediction coefficients, d_j, to be determined,

$$\sum_{j=1}^{M} \Phi_{|j-k|} d_j = \Phi_k \quad (k = 1, \dots M) \tag{3.22}$$

With the values for the d_j's known, equation (3.20) can be used to obtain the discrepancy e_n for the known data set N. If the discrepancies are small, the equation (3.20) can be utilised to predict future data points with a fair degree of confidence.

Equation (3.21) is not necessarily the best means of estimating the covariances Φ_k from the data set. As previously mentioned, MEM and LP are based on the same principal. As a result, Burg's method used for solving the coefficients a_j of MEM can also be used as an alternative for obtaining the linear prediction coefficients d_j.

3.7 Prediction Software

The outlines of a routine that calculates the coefficients $d_1,\ldots.d_m$ (Press et al., 1992) was used by the author to write a 'C' language version of a Linear Prediction Program[5], which basically consists of five modules.

The first module, 'main()', allows the user to input the number of LP coefficients to be used, the number of predictions that will be made, and the number of data points (of the time series) that it needs to read in before making the predictions.

The second module, 'load_array()', reads the actual time series data from a file called 'timeseri.dat' into an array. In the third module 'linpred_coeff()' this information is used to calculate the LP coefficients. As mentioned before, the methodology employed involves a recursive procedure for increasing the order M (the number of coefficients/ poles) by one unit at a time, at each stage re-estimating the linear prediction coefficients d_j, j = 1, .., M.

In the fourth module, 'prediction()', equation (3.20) is used to calculate the linear predictions. This routine references the last M values of the actual data as initial values for the prediction. The last module, 'results_to_file()', does exactly what it's name implies; its function is to write the prediction results to a file with the name 'linpreds.dat' for subsequent user evaluation. The flow diagram of *Figure 3.8* portrays the sequence of events from an overall program viewpoint and serves to illustrate the basic operation behind it.

3.8 Software testing

Three different trigonometric data series (i.e. with periodic fluctuations) were submitted to the program in order to determine whether the MEM/LP software was capable of capturing the underlying pattern and utilise it to make a useful prediction.

The data points for first two series were generated by sinusoidal type linear equations, allowing the characteristics in terms of stationarity, linearity and noise to be known in advance. As such it should be feasible for the algorithm to duly arrive at the model underlying the observed data with a minimum number of poles in characteristic equation (3.18). During the test the predictions had a lead of 1 to 10 values into the future, and the number of LP coefficients varied from 5 to 50. The number of values in the data patterns ranged from 100 to 500.

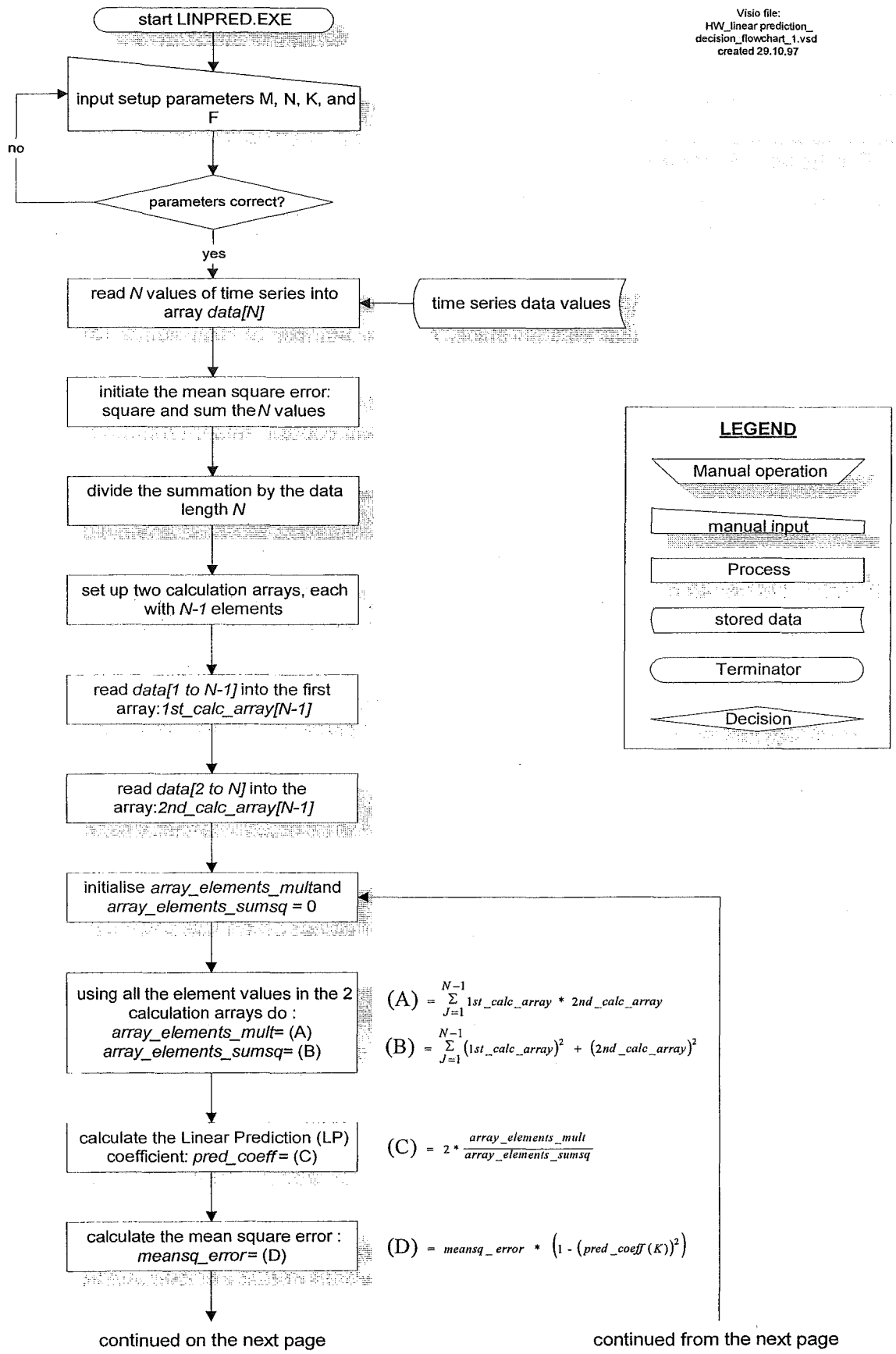
The first set of data points were derived from the linear equation:

$$f(x)_1 = 2\cos x - 8\sin x \tag{3.23}$$

A plot of the data for the first 100 points is shown in *Figure 3.9*

As the MEM/LP software should have no problem in capturing the underlying model the test was focussed on the prediction accuracy versus the amount of historic data available. In the standard auto-regressive (AR) models it is desirable to associate a specific cost function with the forecast errors.

[5] There exist in actual fact two executable files: 'linpred.exe' and 'linear_prediction_program.exe'. They differ only in that the first accepts input data of type **int** and the latter of type **float**.


start LINPRED.EXE
Visio file:
HW_linear prediction_
decision_flowchart_1.vsd
created 29.10.97
input setup parameters M, N, K, and F
no
parameters correct?
yes
read N values of time series into array data[N]
time series data values
initiate the mean square error: square and sum the N values
divide the summation by the data length N
set up two calculation arrays, each with N-1 elements
read data[1 to N-1] into the first array:1st_calc_array[N-1]
read data[2 to N] into the array:2nd_calc_array[N-1]
initialise array_elements_multand array_elements_sumsq = 0
using all the element values in the 2 calculation arrays do : array_elements_mult= (A) array_elements_sumsq= (B)
(A) = Σ 1st_calc_array * 2nd_calc_array
(B) = Σ (1st_calc_array)² + (2nd_calc_array)²
calculate the Linear Prediction (LP) coefficient: pred_coeff = (C)
(C) = 2 * array_elements_mult / array_elements_sumsq
calculate the mean square error : meansq_error= (D)
(D) = meansq_error * (1 - (pred_coeff(K))²)
continued on the next page
continued from the next page
LEGEND
Manual operation
manual input
Process
stored data
Terminator
Decision

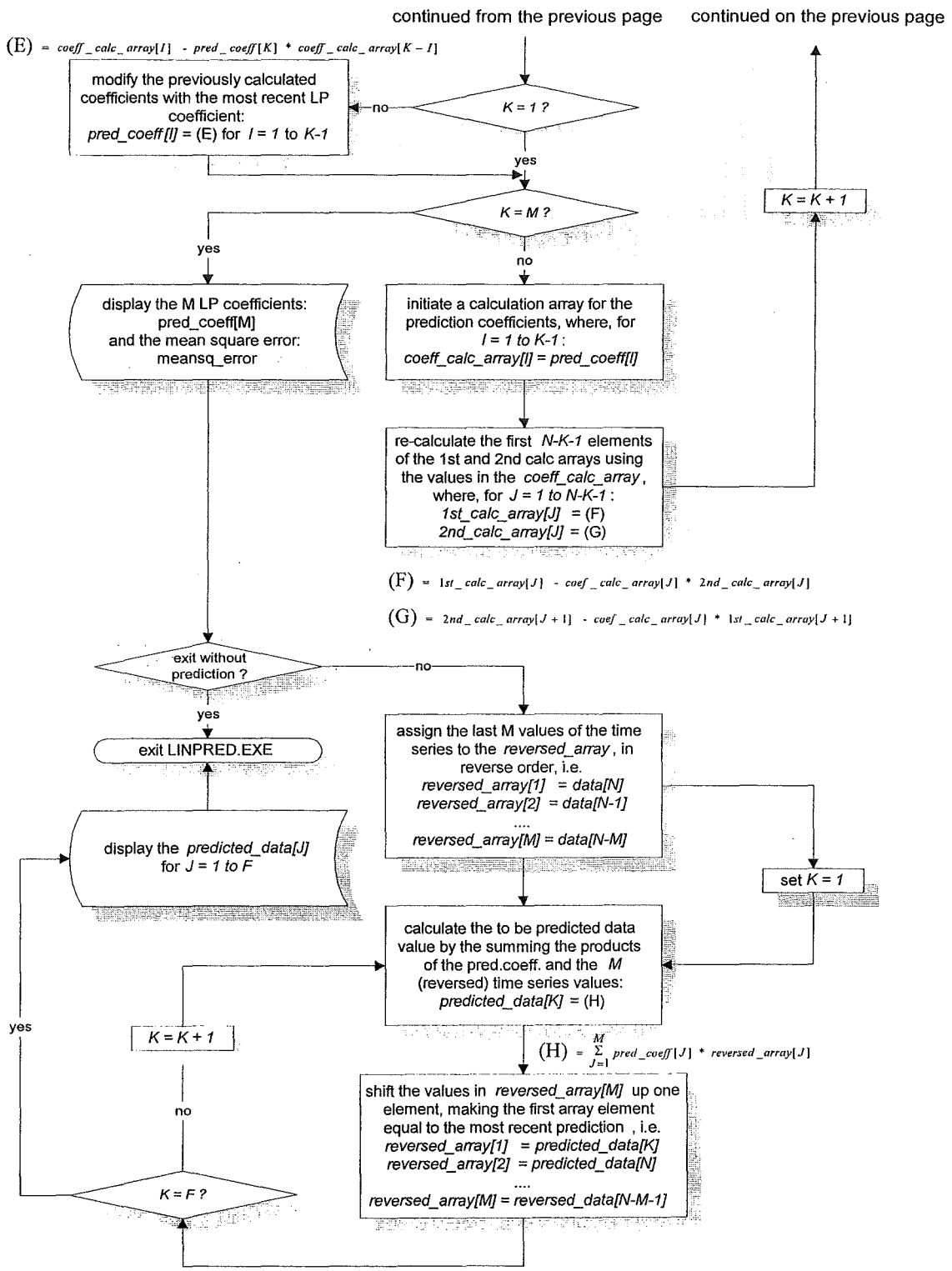


Figure 3.8 **– decision flowchart for the Linear Prediction software**

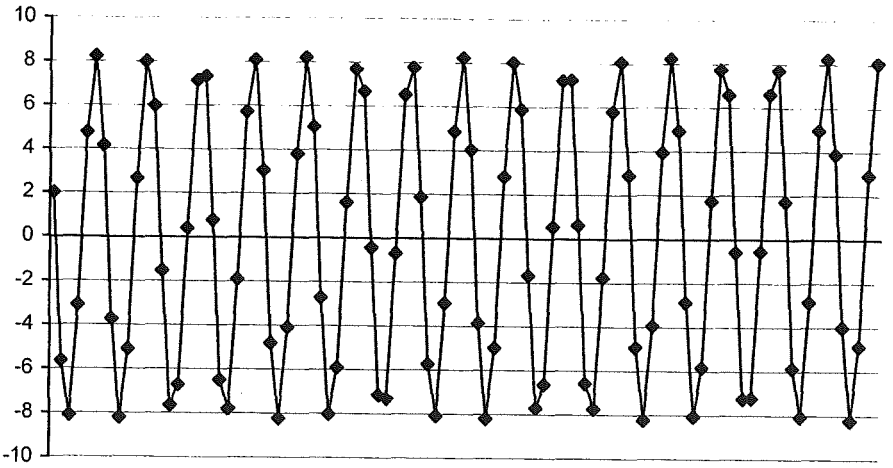


Figure 3.9 – **graph of the function $f(x)_1 = 2 \cos x - 8 \sin x$**

A forecast will be optimal if it minimises the cost (Lütkepohl, 1993). Historic data in sets of 5, 10, 20, ..., up to 490 points, were entered in the algorithm and for each set of data a plot was made of the *mean relative error* (MRE) as defined by equation (3.24).

$$\text{MRE} = \frac{1}{n} \sum_{i=0}^{n-1} \left| \frac{t_i - y_i}{t_i} \right| \tag{3.24}$$

Where t_i is the target (true) value of the prediction for trial presentation i, and y_i is the predicted value obtained by the software. The sum of the errors is divided by n, the number of trials, to get a mean value.

From the test runs it was soon apparent that having just 2 prediction coefficients gave optimum forecasting results. The fact that the equivalent of two poles are sufficient to define the characteristics of the data-model is not surprising given that the original equation has a pair of trigonometric terms. Predictions were made with up to 10 values into the future, using various settings for M, the number of LP coefficients. It was found that regardless of the chosen M, the first value in the prediction series was consistently near to the actual target value. However, the progression after that showed rapidly increasing error, the exception being M=2 as mentioned previously. *Figure 3.10* serves to illustrate this point for an historic data input of 190 values.

The results obtained also confirmed that as the quantity of historic data increases, there is a corresponding decrease in the error associated with the forecasted value. If a set of historic data is used to forecast more than 1 point into the future it can be reasonably assumed that the inaccuracy of the subsequent predictions will increase as it builds error upon error by utilising the last prediction value as a basis for the next. *Figure 3.11* displays the MRE for growing data input, up to a maximum set of 490 points. As expected, predicting 4 values into the future gives a lower MRE than predicting 10 values. As the available historic information increases however, the error reduces quickly until at a set of 290 points the inaccuracy is for all intents and purposes no longer influenced by the number of predictions made. What this means for the software program is that with M=2 and a minimum of around 300 historic points of information it is capable of faithfully modeling the equation $f(x)_1$.

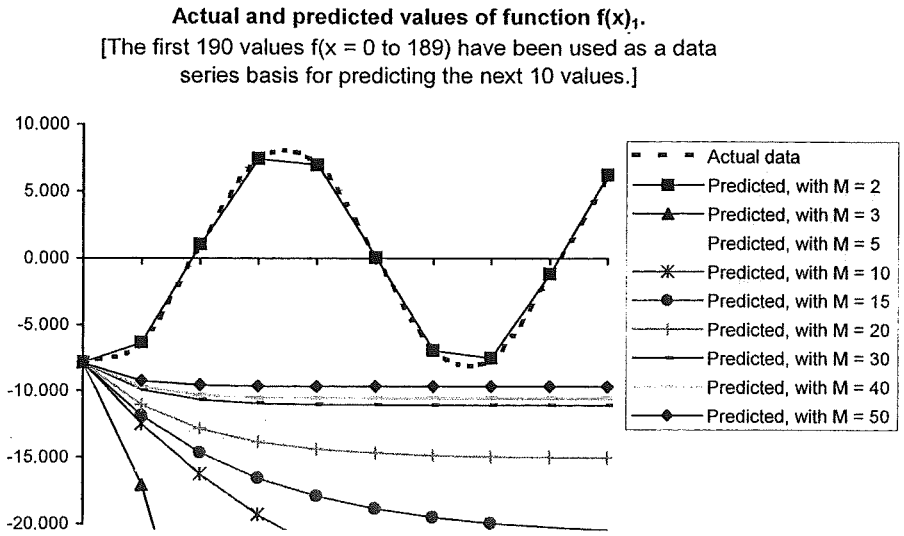


Figure 3.10 – **The effect of M, the prediction coeff., on the prediction values.**

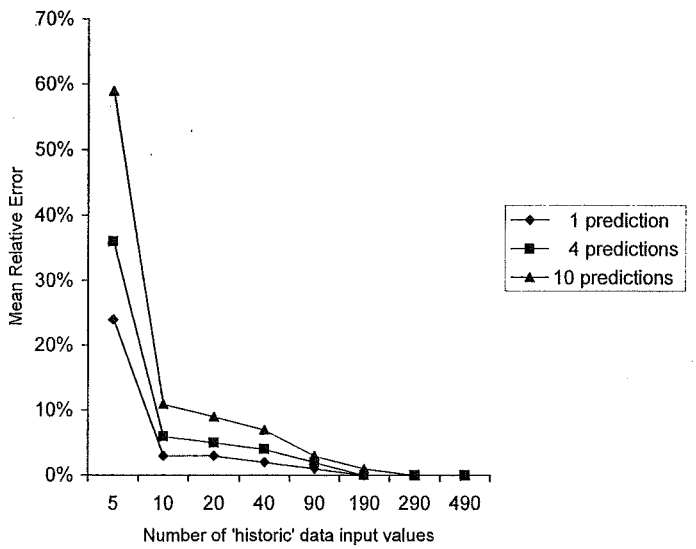


Figure 3.11 – **The reduction in prediction error as the number of available data points increase.**

In order to derive the second set of data points the following linear equation was used:

$$f(x)_2 = e^{\frac{-x}{N}} \sin\left(\frac{2\pi\ x}{50}\right) + e^{\frac{-2x}{N}} \sin\left(\frac{2.2\pi\ x}{50}\right) \qquad (3.25)$$

where N is the number of data points that need to be generated. The equation is the sum of two sine waves with exponentially decaying amplitudes. A plot of the data for the first 190 points is shown in *Figure 3.12*.

The tests with this function were a repeat of that of $f(x)_1$. Again the setting of M=2 for the number of prediction coefficients worked best and gave consistent values for the forecast with a cyclical pattern that closely matched that of the actual function $f(x)_2$. As before, predicting 4 steps ahead gives a lower MRE than predicting 10 values into the future, with the most accurate value given by a single step ahead prediction.

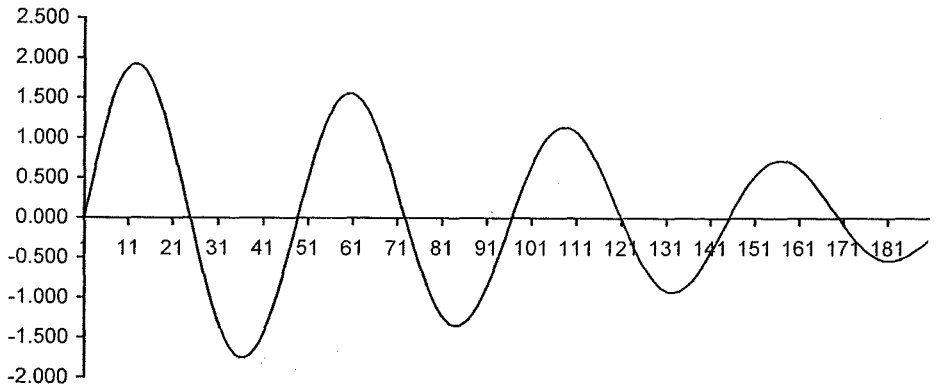


Figure 3.12 – **graph of the function $f(x)_2$.**

It can be seen from

Figure 3.13 that the initial decrease in MRE, the mean relative error, was comparable to that seen for the function $f(x)_1$. The difference is that after the initial dip the errors stay relatively constant and do not decrease in value as was seen previously. The forecasted values appear to have a problem capturing the effect of the steadily decreasing amplitude, although the prediction algorithm has no problem with the wavelength (*Figure 3.14*). That the amplitude remains an issue is shown by the fact that the MRE for multiple prediction stays in the single digit figures despite the extra historic data that becomes available with each new set of data.

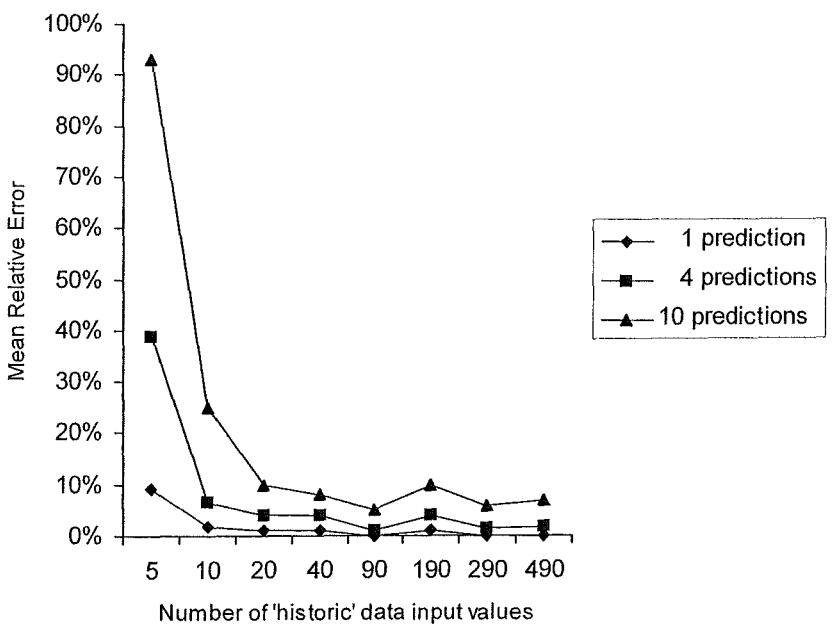


Figure 3.13 - **The reduction in prediction error as the number of available data points increase for function $f(x)_2$.**

The overall larger values for the residuals when making more than a single forecast, although still in the single MRE figures, are an indication that the more challenging aspects of function $f(x)_2$ cannot be faithfully reproduced by the MEM/LP software model.

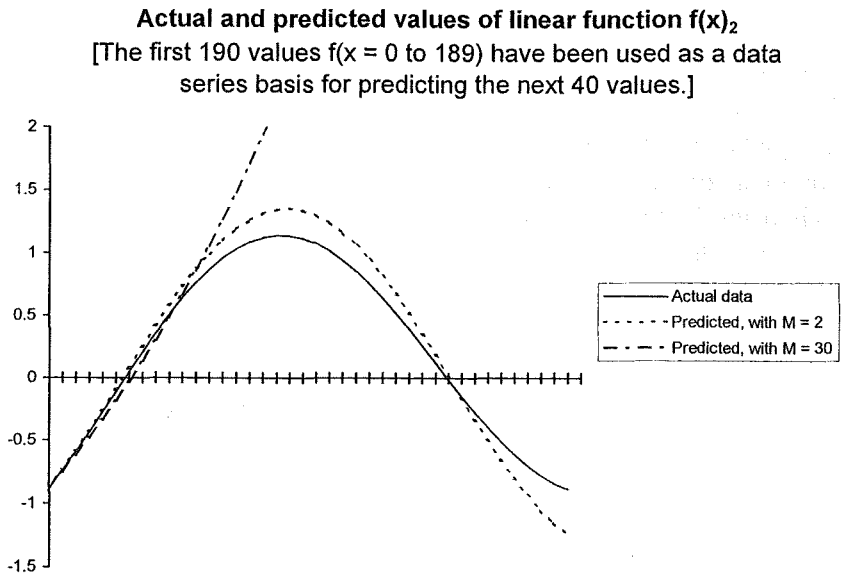


Figure 3.14 – **Actual and predicted values for $f(x)_2$ using 190 past values for input.**

The last test pattern is even more demanding and consists of the real U.S. International Airline passenger data previously shown in *Figure 3.4*. As noted then, this data series has a number of significant features. The first one is that it is a rising pattern that on the average can be described by a linear function of time. Superimposed on this basic trend is a seasonal cycle with increasing amplitude. The seasonal pattern is not exactly a sine wave and indicates the presence of higher harmonics in the waveform.

Despite the fact that domestic hot water demand is unlikely to incorporate an ever-increasing trend as displayed by the U.S. passenger data, this cannot be altogether ruled out in a industrial application; a situation which ideally should be able to covered by the same type of prediction system (the basic goal of a 'black box' EMS).

Having seen in the previous tests that the most accurate forecast is derived from a single step prediction, i.e. the next value (lead = 1) in the discrete time series data, there would appear to be no gain in forecasting further ahead than necessary. This would fit nicely in the scope of a Hot Water Energy Management System (HW-EMS) that concentrates on predicting a single daily figure of hot water use (or its equivalent in terms of energy). Nevertheless, it is interesting to see if the quality of the model is maintained when faced with multiple step ahead predictions. If provision is made for a HW-EMS that needs to forecast a number of steps; for instance in order to establish a hot water usage *profile* for the next day, then the lesser accuracy would have to be taken into account. This is an issue that will arise when electricity spot pricing is incorporated in the system.

While focussing on finding a value for M which results in the residuals error being as small as possible, the data from the international airline passengers was used to make predictions of one and four months into the future. An additional variation that was introduced c.f. the previous two test functions was the concept of the moving data window. The width of the window is equal to 24 months of data, with the forecasted figures being the number of passengers in the subsequent months. The aim here is to keep the data input 'local', thus hoping to avoid any non-stationary characteristics. It is also interesting to evaluate the residual error when the passenger series has been de-trended; theoretically allowing the MEM/LP software to concentrate on the more pertinent characteristics of the data.

The results obtained with larger and larger data sets being made available to the software in steps of 12, 24, 36 ..., 120 months are displayed in *Figure 3.15*. As for previous trials the error is smallest with a single month forecast and increases as the number of predicted values increase with leads from 4 and 10 months. This smallest error results using two prediction coefficients (M equals 2) but as the lead time grows it turns out that M equal to 15 on the whole returns the best forecasted values. Unexpected is the increased error across the range of predictions when the historic data input is at its maximum value of 120; it points to the fact that the real-life data incorporates relevant characteristics that the model has not extracted. Pre-processing the data or even choosing a different model might be a further requirement.

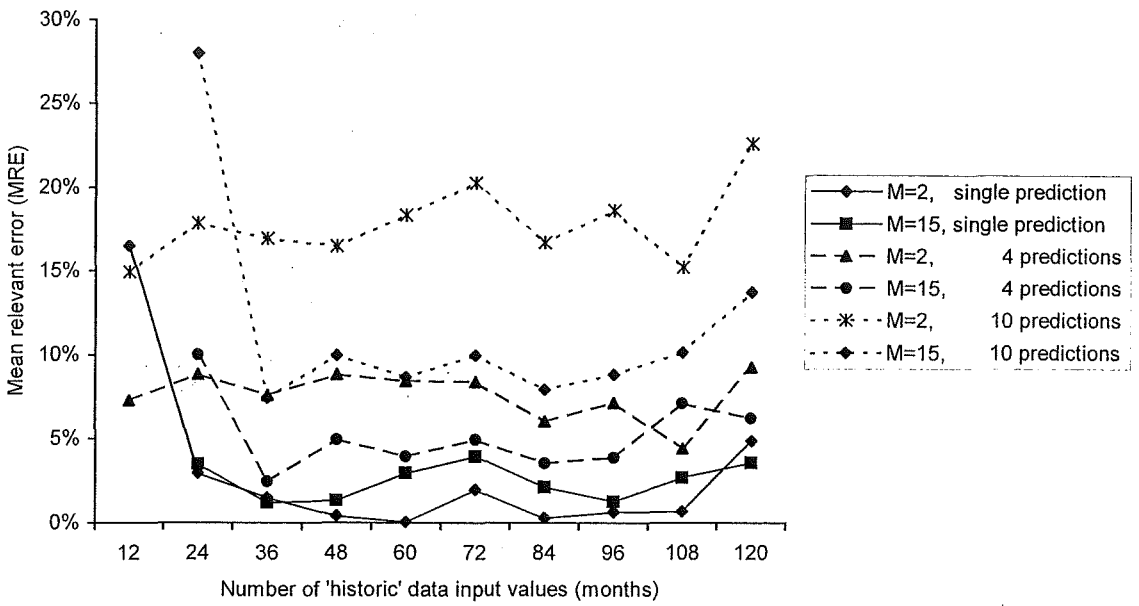


Figure 3.15 – **Residual errors for increasing amounts of past data (U.S. international airline passengers).**

Pre-processing is the most obvious option to implement first, and when the passenger data has been de-trended using a least squares derived straight line a better result for the error was obtained. *Figure 3.16* displays a marked reduction for forecasting error with LP coefficient M = 2; it has less influence on M = 15 until the lead time reaches 10 months. Further fine-tuning by filtering the seasonal component, choosing optimal data segments, and/or 'massaging' the linear prediction coefficients (a technique described in Press et al.) all offer the potential for additional improvement in the outcome.

The notion of trials with a moving window was born out of the desire to see the result of a minimum amount of historic data being used for forecasting in conjunction with minimising the non-stationary traits of the data input series. The fixed width of 24 months was not an arbitrary choice but based on the size of the segment of data needed to produce the first single figure error with a one step prediction, as shown in *Figure 3.15*. A plot of the actual data and the forecasted passenger figures is given by *Figure 3.17*. When checked for a range of LP coefficients it turned out that M = 23 returned the most accurate single step forecasts for the moving data window; this in contrast to smaller residual errors obtained with M = 2 and no window.

However, when additional predictions are made further into the future the value of M = 23 rapidly degrades the resultant accuracy and M = 2 is once again the preferred choice (see *Figure 3.18*).

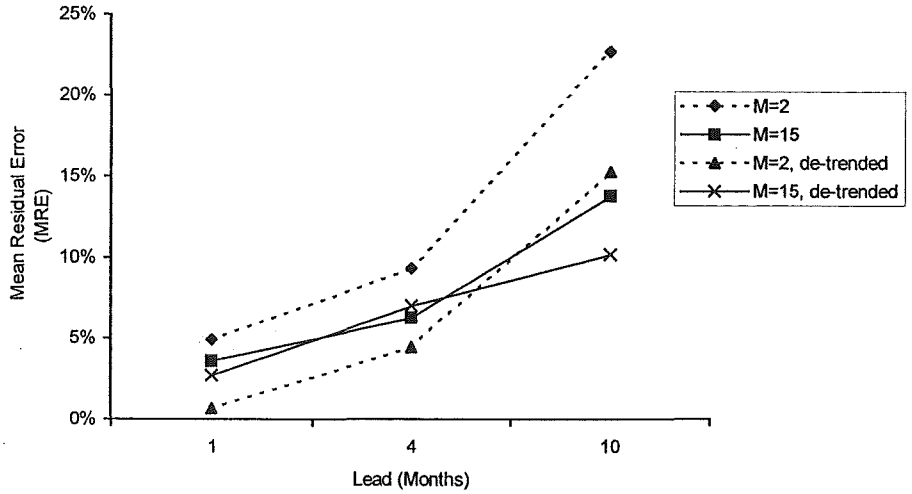


Figure 3.16 – **The effect of de-trending the U.S. passenger data on subsequent predictions for 1, 4, and 10 months ahead (120 historic values used for input).**

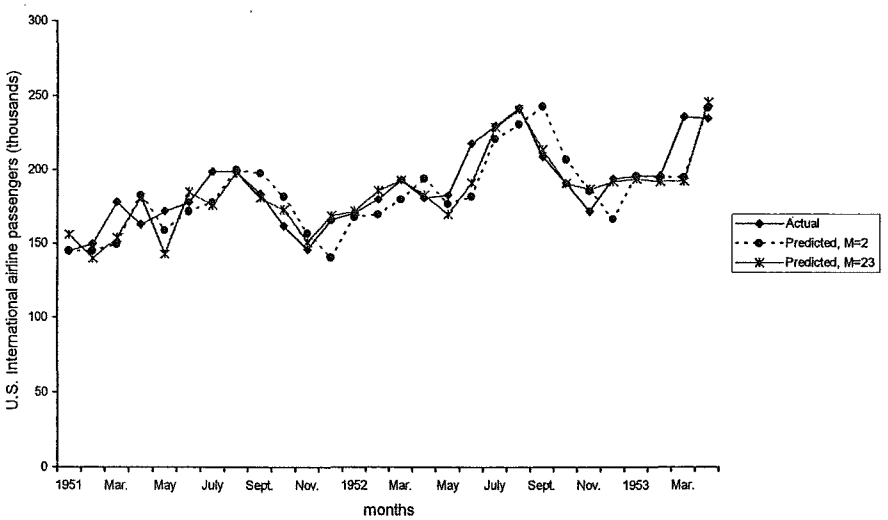


Figure 3.17 – **Single predictions made with a moving 24-month wide window of data input.**

The stage of the research that had now been reached did not allow for testing of *actual hot water demand* data; this information was simply not available and was in the process of being obtained by a test rig situated in domestic household.

Of course, at the end stage of thesis write-up these results were available and it would have been possible to subsequently test them. However, there seems to be little point in pursuing such a test, as the result obtained by the three trial cases clearly indicate that some form of *manual* selection (no. of coefficients, size of data segments) is needed to obtain optimal results. The important ability to *adapt* to different domestic and industrial environments is compromised in this manner. Thus there are inherent difficulties with utilising the spectral model in a *fully automated, stand-alone* Energy Management System. (i.e. different choices of M, optimal data sizes, filtering of data and other fine-tuning techniques all require *manual* input; this is *not* an option in an automated system)

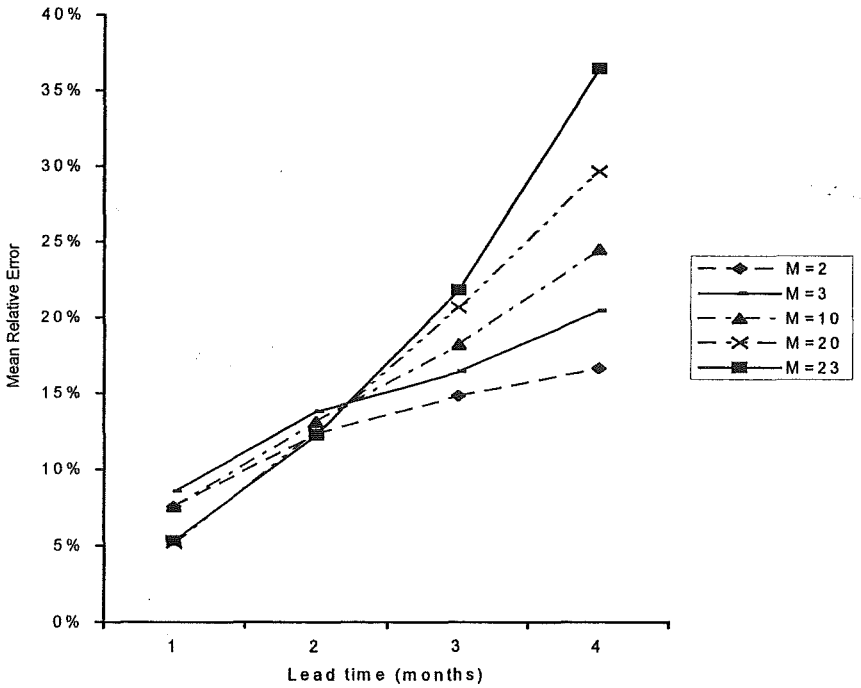


Figure 3.18 - **Forecasting accuracy with increasing lead times, using a 24-month wide moving window (U.S. International passengers).**

3.9 Conclusions / Discussion

This chapter focused on acquiring a suitable means for forecasting the future energy demand for a domestic hot water heating cylinder. Ideally any prospective forecasting procedure for an eventual EMS would fulfil five requirements:

- Self-initialisation and automation.
- Accuracy
- Adaptiveness
- Comprehensibility
- Computational economy

The classical methods of time series prediction were examined for applicability at some length, as this area is mature with established and well-understood techniques. An often-quoted statement in a majority of the literature is that the care taken in the selection of a proper prediction model means the difference between a reliable forecast and a questionable forecast.

The linear prediction technique chosen incorporated a number of features thought to aid the successful modelling of the hot water demand time series. MEM, Maximum Entropy Method, also known as the all-poles model, is an auto-regressive based method of forecasting where old values of the variable are used for prediction (in statistical terms, regression is a standard technique by which one or more *independent* variables are used to predict a single *dependent* variable). MEM characterises a known time series in terms of a finite number of poles that best represent its spectrum in the complex z-plane. This characterisation can then be used with linear prediction to extrapolate the series in terms of

the same poles. The poles allow a better representation of a time series power spectrum that possesses sharp, discrete peaks. An MEM 'C' routine adapted from a number of numerical method programs was then tested for prediction accuracy.

Test patterns generated by trigonometric functions and real-life U.S. airline passenger statistics provided the trial data for the prediction program. The program handled cyclic data well and proved to be fast in data-processing. However, the correct modelling of the pattern oscillations proved sensitive to the initial number of poles chosen. The 'M' parameters that worked fine for one set of trigonometric data failed to recover a reliable model for another. The inherent frequencies in a particular time series pattern could only be modelled by a limited number of poles. The introduction of trend, seasonal influence and noise as present in the actual airline passenger data effected the prediction accuracy and showed that the quality of the model-algorithm was not robust in a general sense (i.e. different values of the LP coefficient and varying data segments produced diverse results). Predicting more than one step into the future also proved to reduce the accuracy of the forecast and limits the usefulness of the software in providing a multiple prediction profile.

As there is no guarantee that a household or industrial process will keep to the same regular demand pattern in-perpetuum, the tentative conclusion would have to be that the combination MEM and linear prediction are not well suited to a *fully-automated black box* approach. In particular they lack the elementary adaptability and robustness needed for the visualised EMS and are not particularly well suited to an unfamiliar dynamic environment. It is envisaged that only in an industrial situation for fluid heating, where known alterations in patterns are limited, the linear prediction program would prove useful and adequate.

The notion that the program needs to be able to adapt to a relatively unknown dynamic environment is reinforced if the variant factors that can affect hot water use are taken into consideration. Some of these factors are:

- *Occupancy* - Variations in the number of people living in houses will be a major influence on individual hot water demands. This is one of the more salient points that the MEM/LP prediction algorithm is unable to adapt to. For each domestic situation a judicious choice of the various parameters is needed to accomplish a reliable forecast.
- *Juvenile/mature consumer demand* – Younger members of a family tend to change their hygiene patterns as they progress from child to teenager or adult, whereas the older members could possibly become infirm. All members could fall ill for a longer period of time and thus reduce/increase hot water demand dramatically.
- *Seasonal variation* - The activities of the individual household members can change during the different times of the year and alter their hygiene pattern (e.g. participation in different summer/winter sports).
- *Utilisation pattern* - The pattern of hot water use for a given period can be quite dissimilar from one household to the next, even if they have the same appliances and an equivalent family composition.
- *Heat dissipation* - As the water can be stored for a considerable length of time, with or without intermediate heating, the rate of heat loss is going to be determined by the average water temperature, the surface area, the cylinder insulation thickness and the ambient temperature.
- *Loft temperature* - If the hot water cylinder is installed in the loft or attic, especially an uninsulated one, then its surrounding temperature can vary greatly, both as a daily and seasonal cycle i.e. sunny versus overcast, summer versus winter. This will affect heat loss

to a certain degree. Fortunately for the more modern cylinders, fitted with better, thicker, insulation, the daily temperature fluctuations should have a reduced influence. The prediction algorithm itself takes care of the seasonal effects by virtue of the moving data window.

- *Cold inlet water temperature* - The temperature of the replacing cold water varies much the same as the loft temperature, i.e. on a seasonal basis. Again though, the moving data window will cause the forecast mechanism to adapt to these slow changes. Only severe, almost random, weather influences, like snap-frosts or heat waves, should negatively affect the predicted value.
- *Cylinder design* - Different countries with different manufacturers and having no set standard will mean that the design and make-up of the cylinders can vary greatly. For instance, the presence and shape off internal baffle plates installed over the cold water inlet will directly affect the layering and internal turbulence.
- *Cylinder capacity* - Different sizes of hot water storage cylinders are available on the market, i.e. 120, 180 and 270 litres.
- *Appliances* - The type and number of appliances installed in a household i.e. washing machines and dishwashers, will have a significant impact on the hot water demand if they are hooked up to the hot water supply.
- *Plumbing* - The lengths of pipe, and even the diameter, which run from the hot water cylinder to the various user points can vary considerably from one household to another. The amount and quality of the insulation around these pipes to prevent further heat loss will also affect the hot water demand.
- *User points* - The number of wash-basins, sinks and bathrooms vary from house to house have an effect on standing losses as well as overall usage.
- *Household income* - This will strongly influence the storage temperature, quality of insulation, and the amount of water that literally goes down the drain.
- *Geographical location* – Hot versus cold climates should see a pattern difference.

3.10 Summary

In support of the standard statistical techniques, and the MEM/LP algorithm in particular, it should be noted that it is of course possible to refine the software model further. One way is that it will first collect a certain amount of data (much as the neural based program of *Chapter* 8) and subsequently choose the appropriate parameter values by recursive means, minimising the prediction error in the process. Even the data could be pre-processed to filter out spurious noise; although this carries the inherent risk that pertinent information is ignored. The hot water demand data specifically could be regressed with such explanatory variables as heating and cooling days, electricity prices, income per capita and perhaps seasonal variables or time trends.

However, the fact remains that it lacks the necessary flexibility required for the desired FEMS. The quality of the forecast is greatly affected by the chosen model and that it is a linear predictor only (i.e. any linear combination of inputs produces the same combination of linear output components). And if there exists the possibility that non-linear factors influence the hot water demand than there are more suitable alternatives available; methods that will adequately cope with both linear/non-linear situations and at the same time are by their very nature dynamic and adaptable.

The aim of this research is to produce a 'black box' Fluid Energy Management System which does not need manual fine-tuning, and that effectively deals with dynamic electricity spot prices as well as hot water energy demand. This, with multivariate input, represents in all probability a non-linear mechanism, and there would appear to be little benefit in pursuing the MEM/LP combination for forecasting.

A possible alternative is offered by the so-called *artificial neural network*, a model touted by Masters (1994) as being highly effective when applied to pattern recognition. When tested by Varfis et al. (1990) on a non-stationary forecasting problem with a uni-variate time series the conclusion was that good accuracy was obtained c.f. the Box-Jenkins method. The low requirements of statistical knowledge and of pre-processing (the neural network coped with the non-stationary features of seasonal behaviour and rising trend without having to resort to transforming the input data) counterbalanced the slightly more complex implementation and increased computation. The statistician's adage that many neural network models are of strictly theoretical importance, too slow to be practical, good training algorithms may not exist, too much computer memory may be required, or their performance in real-life problems leave much to be desired, is rapidly being negated as more and more models are proving to be immensely valuable. Tasks that were formerly performed by statistical techniques like discriminant analysis can now be done faster and more effectively by neural nets. That there are limitations in term of prediction is indicated by Ginzberg et al. (1991) whom, having found that ANNs can be trained to learn the time series of a dynamical system and be used to predict the next value of a given series, concluded that a network can discover the correct law if its architecture can accommodate it; otherwise it provides an approximation whose accuracy deteriorates quickly in long term prediction.

It is expected that, aside from varying electricity tariffs, additional variables such as ambient temperature, day of the week, and holidays exert some degree of (non-linear) influence on the hot water demand in a household. Neural networks will accommodate without commotion these additional parameters as well as any number of past time series data. Another advantage is that neural network models exist that optimise the information storage of past *temporal* data in the synaptic weights, thus supporting the 'moving window' concept as well as eliminating the need of keeping in memory *all* the historic data.

Chapter 4. Artificial Neural Networks

4.1 Introduction

Neural networks are *adaptive parallel processing systems* inspired by the anatomy and physiology of the brain. Artificial neural networks (ANN) as used for engineering purposes are interconnected networks of simple processing elements. Communication between the processing elements occurs along paths of *variable connection strengths*. This chapter introduces the fundamental concepts of neural networks before examining a number of different neural networks - the *Hopfield* network, the *Self-Organising Map* network, the *Back-propagation* network and the *Radial basis* network. Hardware implementations of artificial neural networks are then briefly examined and the chapter concludes with a brief discussion of the similarities and differences between statistical methods and artificial neural networks. The aim of the chapter is to familiarise the reader with general ANN terminology and functionality, and to show the maturity of this recent arrival in applied engineering.

4.2 The history of artificial neural networks

Work on artificial neural networks, commonly referred to as "neural networks", has been motivated right from its inception by the recognition that the brain computes in an entirely different way from the conventional digital computer. The struggle to understand the brain owes much to pioneering work performed at the beginning of the 20^{th} century, when the idea was introduced of neurons being structural constituents of the brain. Typically, neurons are five to six orders of magnitude slower than silicon logic gates; events in a silicon chip happen in the nanosecond range, whereas neural events happen in the millisecond range. However, the brain makes up for the relatively slow rate of operation of a neuron by having a truly staggering number of neurons (nerve cells) with massive interconnections between them; it is estimated that there must be on the order of 10 billion neurons in the human cortex, and 60 trillion synapses or connections (Shepherd et al., 1990). The net result is that the brain is an enormously efficient structure. Specifically, the *energetic efficiency* of the brain is approximately 10^{-16} joules (J) per operation per second, whereas the corresponding value for an early model computer is about 10^{-6} joules per operation per second (Faggin, 1991).

The brain is a highly complex, non-linear, and parallel computer (information processing system). It has the capability of organising neurons so as to perform certain computations (e.g. pattern recognition, perception, and motor control) considerably faster than a modern digital computer.

The computer on the other hand, is far better than the human brain at arithmetic and formal logic. As such, a single computer architecture may not be enough to solve all problems. Neural networks have developed as information-processing systems with architectures inspired by the brain. In its most general form, a neural network is a machine that is designed to *model* the way in which the brain performs a particular task or function of interest; the network is usually implemented using electronic components or simulated in software on a digital computer. The latter is more common due to the wide proliferation of relatively inexpensive personal computers and readily available software.

The field of neural networks grew from collaboration between engineers attempting to find alternative solutions to complex problems, and neurobiologists interested in understanding how intelligence emerges from the interaction of many neurons in the brain. As a result of

greater emphasis being placed on either biological plausibility or technological achievement, two subfields of neural networks have emerged - biological modelling and engineering application. Neurobiologists look to (artificial) neural networks as a research tool for the interpretation of neurobiological phenomena. For example, neural networks have been used to provide insight on the development of pre-motor circuits in the oculo-motor system (responsible for eye movements) and the manner in which they process signals (Robinson, 1992). On the other hand, engineers look to neurobiology for new ideas to solve problems more complex than those based on conventional hard-wired design techniques. For instance, Hummels et al. (1995) developed a locally optimum signal detector to adaptively detect small sinusoidal signals in the presence of noise. The motivation is to develop a new receiver that is superior to one designed by conventional methods. The use of brain inspired models can also be expressed in yet another way; Andreou (1992) uses the neurobiological analogy to influence the design of analog VLSI models.

The artificial neural networks discussed in this chapter have, for the most part, been developed with technological applications in mind. The following historic excerpt is from Dingle (1992).

The study of neural networks really began with McCulloch and Pitts (1943) who attempted to understand what the brain might actually be doing. By modelling neurons as binary threshold devices, they showed that networks of simple elements could have immense computational power; specifically, they demonstrated such networks could realise any finite logical expression.

McCulloch and Pitts had shown how networks of neuron-like elements could compute. The next problem facing researchers was to understand how such networks could learn. In 1949, Donald O. Hebb provided the first explicit statement of a physiological learning rule:

> *"When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."*

This, Hebb's *postulate of learning*, is the oldest and most famous of all learning rules (Hebb, 1949). Hebb proposed this change as a basis of associative learning (at the cellular level), which would result in an enduring modification in the activity pattern of a spatially distributed "assembly of nerve cells". Despite the non-mathematical nature of the learning law, it has formed the basis of many learning algorithms that modify the strength of synaptic coupling between neuron-like elements as a function of pre-synaptic and post-synaptic activity.

The *perceptron* (Rosenblatt, 1958) was the first true artificial neural network. It was computationally feasible on the hardware of that time, was based on biological models, and was capable of learning. Rosenblatt presented an algorithm by which his perceptron could be trained. He proved that if a set of training patterns is learnable by the perceptron, his algorithm is *guaranteed* to converge to a set of network weights (the value of the connection between individual neurons) that enable correct response to the training set. This theorem, along with some fairly impressive early demonstrations of problem solving, propelled neural network research forward for a short time. Unfortunately, the perceptron also suffered from a significant weakness. It is only capable of solving classification problems that are *linearly seperable* at the output layer. Fate then conspired against artificial neural networks. Rosenblatt's former schoolmate, Marvin Minsky, along with Seymour Papert, published the book *Perceptrons* (Minsky et al., 1969), which went to great lengths expounding on the

weaknesses of the perceptron model (e.g. it's inability to solve the 'exclusive OR' problem). Because of the book's mathematical rigour, and also because the authors were very well known and respected researchers in artificial intelligence, a shadow was cast on neural network research. Soon after, in 1971, Frank Rosenblatt died in a boating accident. Without his support, and in the wake of *Perceptrons*, money for neural network research rapidly dwindled. The direction of artificial intelligence research shifted to sequential symbolic processing.

Although some neural network research did continue in the ensuing years, it was at a considerably slowed pace and was primarily directed towards biological modelling. A few pioneers developed much interesting theory but having little practical success (Anderson et al., 1988), (Rumelhart et al., 1986). However, with the 1980s came a revival of interest in artificial neural networks and a considerable number of important discoveries were made. In 1982 John Hopfield published a neural network that behaved as *content-addressable memory*, correctly recreating an entire memory from any subpart of sufficient size (Hopfield, 1982). In the same year, Teuvo Kohonen (1982) introduced a network, the *self-organising map*, which was able to discover important features in a set of input patterns and spatially order them to form a topographically organised map. However, it wasn't until 1986, when David Rumelhart, Geoffrey Hinton, and Ronald Williams published "Learning Internal Representations by Error Propagation" that research into artificial neural networks once again began to receive significant recognition and funding. Their development of a multi-layer feedforward network that was not restricted to linearly separable training sets, along with a reasonably effective training algorithm for it, demonstrated that artificial neural networks could provide real solutions to practical problems.

4.3 Properties of artificial neural networks

Until recently, information processing has involved devising algorithms or rules to solve a problem and encoding them in software. Neural networks provide a radically different approach to information processing. Based upon modern neurophysiology, neural networks are made up of many neural units (models of neurons) which interact with each other through weighted connections. Neural units tend to be very simple and, in isolation, have extremely limited computational power. The information processing capability of neural networks is a collective phenomenon resulting from the interaction of many neural units. The collective properties of neural networks tend to be relatively insensitive to the detailed operation of neural units.

Neural networks are able to learn from experience by modifying the strengths of connections between neural units. In this way, knowledge of a particular pattern becomes distributed over the connections among a large number of neural units. The patterns themselves are not stored, rather the weights of the network becomes such that the patterns can be recreated when required. Consequently, the time for a neural network to respond to a given input pattern is independent of the number of memories it contains.

Neural networks do not need to be trained on all possible input patterns because they are able to *generalise* from a set of typical examples. In other words, after appropriate training, neural networks are able to respond correctly to input patterns not previously encountered. For example, a neural network trained to recognise characters of the alphabet is still able to classify input characters correctly when they are corrupted by significant amounts of noise. Neural networks are also able to cope with incomplete input data and even with partially incorrect data.

Because information is distributed throughout the neural network, failure of individual neural units or their connections is not catastrophic, rather the performance of the network deteriorates gradually as more component fail. This is an important property of neural networks and is known as *graceful degradation.*

4.4 Components of artificial neural networks

A neural unit is a highly simplified model of a neuron. Many neural units are interconnected to form a neural network. Neural units interact with each other and with the environment through uni-directional weighted connections. The architecture of a neural network describes which neural units are interconnected and how the network interacts with the environment. The strengths of the connections between neural units are termed *weights* and represent the synaptic coupling between neurons. In general, the weights may be positive or negative corresponding to excitatory and inhibitory synapses respectively. The knowledge of a neural network is embodied in the weights between neural units and, therefore, learning is a matter of searching for a set of weights that produces the desired network behaviour. A *learning algorithm* provides the mechanism for finding an appropriate set weights.

A neural network can be specified by its architecture, its neural units and its learning algorithm. Although each is discussed separately here, they are not independent considerations, as is revealed by examination of four neural networks in *Sections* 4.6.1 to 4.6.4.

4.4.1 Network architecture

The *architecture* of a neural network describes the connections between neural units. The strength of the connection from unit i to unit j is represented by the weight w_{ji}; if no such connection exists then $w_{ji} = 0$. Therefore, the network architecture specifies which of the weights w_{ji} can be non-zero.

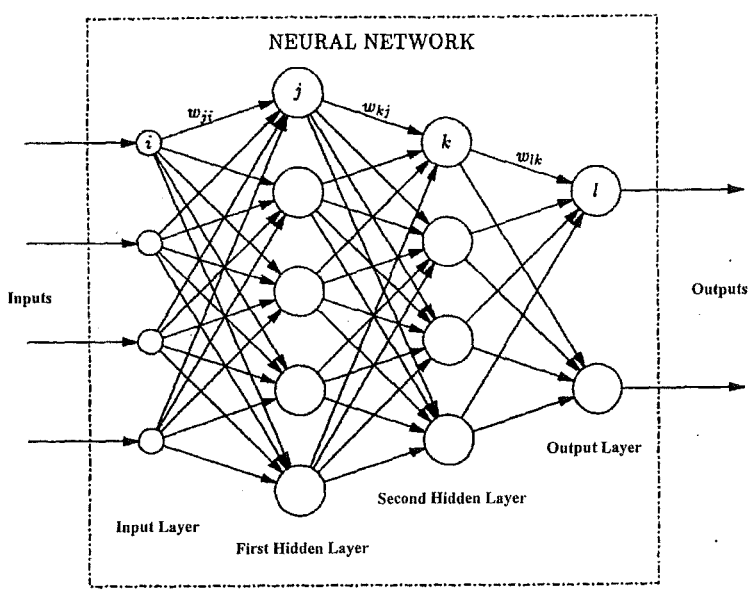


***Figure 4.1*– A four layer neural network with two hidden layers.**

A neural network can be thought of as one or more *layers* of neural units. The *input layer* consists of a number of fan-out elements that distribute the external inputs to the neural units of the following layer. The final or *output layer* contains the neural units whose outputs are

available to the environment. Between the input and output layers there may be any number of *hidden layers*, so named because they are effectively hidden from the environment. When describing the architecture of a neural network the number of layers is often quoted. However, in literature there is some disagreement as to whether the input layer should be counted. The reason for this is that the input layer merely distributes the inputs rather than processing them. In this thesis, the number of layers refers to the layers of neural units plus the input layer. Therefore, the network of *Figure 4.1* is described as a four layer network.

Having detailed the arrangement of neural units within a neural network, it is necessary to specify the connections between them. Connections are described as (a) *feedforward* if they link neural units of one layer with those of the following layer, (b) *feedback* if they join neural units in one layer with those of the previous layer, or (c) *lateral* if they connect neural units within a layer. *Figure 4.1* shows a four-layer network with feedforward connections of consecutive layers, which is the typical architecture of back-propagation neural networks (section 4.6.3).

Many other neural network architectures are possible. For example, the self-organising map is a single layer network where each neural unit receives the entire input pattern. Similarly, the Hopfield network (section 4.6.1) is a single layer network but with a full complement of lateral connections. The Hopfield is also described as *fully connected*, because each neural unit interacts with all other neural units.

4.4.2 Neural units

Neural units are models of neurons and, as such, should be as simple as possible while retaining the features necessary for information processing. In general, a neural unit j receives several, say M, inputs x_i through weighted connections w_{ji} and from these derives a single input y_j (*Figure 4.2*), which then becomes the input x_i to other neural units. A neural unit usually has a characteristic threshold value θ_j which is subtracted from the linear sum of the inputs. A typical neural unit can be expressed mathematically as :

$$y_j = f\left[\sum_{i=1}^{M} w_{ji}\, x_i - \theta_j\right] \tag{4.1}$$

where $f[\cdot]$ is the transfer function or the neural unit. It is often convenient to represent the threshold as a weight from an extra input x_0 :

$$y_j = f\left[\sum_{i=0}^{M} w_{ji}\, x_i\right] \tag{4.2}$$

where $x_0 = 1.0$ and $w_{j0} = -\theta_j$. The advantage of this representation is that an appropriate threshold value can be learnt along with the strength of connections between neural units.

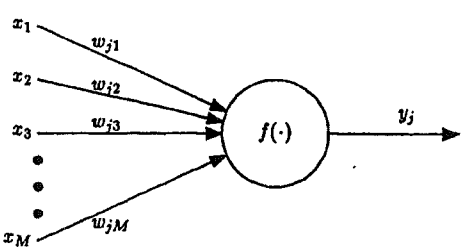


***Figure 4.2* – A neural unit receives inputs x_i through weighted connections w_j and produces a single output y_j. The transfer function f [·] depends on the neuron features being modelled.**

The form of the transfer function $f[\cdot]$ depends on the particular features of a neuron that are modelled. Early neuron models, including the original perceptron, used a simple threshold function. If the weighted sum of inputs is less than the threshold, the neuron output is 0. Otherwise the output is 1 (*Figure 4.3a*). In some models, such as versions of the perceptron, the output would be the weighted sum itself when the threshold is exceeded (*Figure 4.4b*). The range of the activation function (values it can output) is usually limited. The most common limits 0 to +1, while some range from -1 to +1.

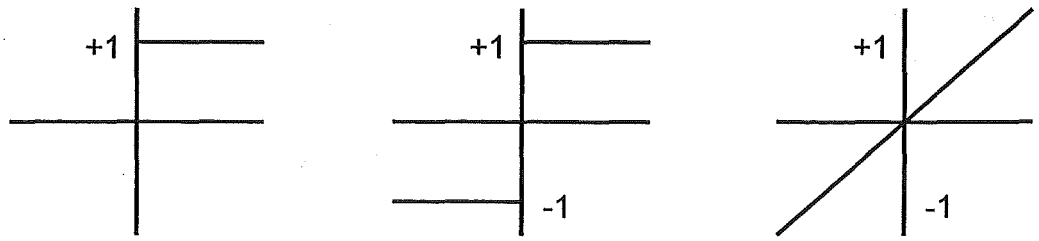


Figure 4.3 **- Typical transfer functions for neural units: (a) unit step function, (b) signum function, (c) linear function.**

The majority of current models use a *sigmoid* (S-shaped) activation function. A sigmoid function may be loosely defined as a continuous, real-valued function whose domain is the real numbers, whose derivative is always positive and whose range is bounded. The most commonly employed sigmoid function is the *logistic* function.

$$f(x) = 1/(1 + e^{-x}) \tag{4.3}$$

One advantage of this function is that its derivative is easily found:

$$f'(x) = f(x)\,(1 - f(x)) \tag{4.4}$$

Other sigmoid functions, such as the *hyperbolic tangent* and (scaled) *arctangent*, are sometimes used.

$$\text{Tanh}\,[x] = (e^{x} - e^{-x}) / (e^{x} + e^{-x}) \tag{4.5}$$

In most cases, it has been found that the exact shape of the function has little effect on the ultimate power of the network, though it can have a significant impact on the training speed.

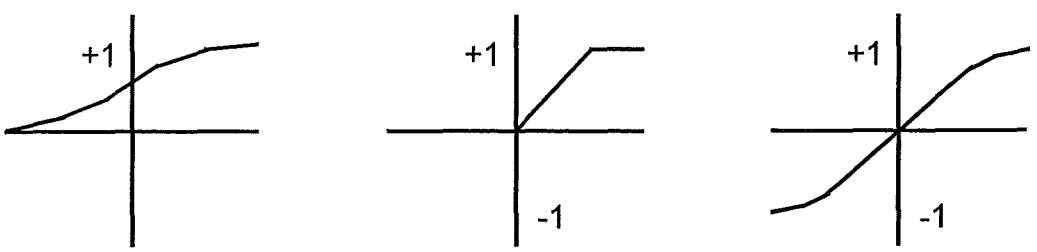


Figure 4.4 **- Typical transfer functions for neural units : (a) sigmoid function, (b) linear function, (c) tanh function.**

Kenue (1991) reports that a relatively small derivative of the logistic activation function slows learning in the basic backpropagation algorithm. Kalman et al. (1992) make a very eloquent case for choosing the hyperbolic tangent function.

It should be noted that sigmoid functions never reach their theoretical minimum or maximum. For example, neurons that use the logistic function should be considered fully activated at around 0.9 and turned off at about 0.1. It is certainly reasonable to use the

extremes of 0.0 and 1.0 as inputs to a network. It is futile, though, to attempt to train a network to achieve extreme values as its output (Masters, 1994).

Although a linear function is sometimes employed (*Figure 4.3c*), non-linear functions are far more common because they significantly increase the collective computational power of the neural network.

4.4.3 Learning algorithms

Neural networks are able to learn from experience by appropriately adjusting the strengths of connections or weights between neural units. A *learning algorithm* provides the mechanism for modifying the weights of a network based upon a number of typical examples, known as the *training set*. The training set and the learning algorithm determine what the network learns and how well it generalises to patterns not previously encountered. Once trained, the performance of the network is evaluated by presenting it with a number of novel inputs patterns, known as the *test set*, and calculating a performance measure (e.g. number of correct responses, mean squared error, etc; as for the statistical methods).

Learning algorithms modify the weights between neural units in accordance with the input pattern and the network's response to it. These algorithms form a spectrum, at one end of which is learning with an error-correcting teacher and at the other is completely spontaneous unsupervised discovery. In between is a continuum of rules including a number of graded learning schemes where the neural network is given an indication of its performance.

Learning with an error-correcting teacher or *supervised learning* enables a neural network to learn arbitrary associations between input and output patterns. The network is presented with an input pattern and the corresponding target output pattern. The weights between neural units are then modified to reduce the error between the target pattern and the output pattern produced by the network.

Graded learning algorithms are usually less capable and less generally applicable than the supervised schemes. Their main advantage is that it is not necessary to know the correct output pattern. Instead of being given the target output for each input pattern, the network receives only an indication of its performance after several training patterns. This is usually a score or grade that represents the value of some performance measure or cost function. Graded learning is particularly applicable to control problems where there is no way of knowing the correct outputs, for example, balancing a broomstick on its end. In this case the performance measure may be a binary value corresponding to success or failure (Barto et al., 1983), or it may be the sum of the absolute angular deviation of the broomstick from the vertical.

Unsupervised learning is also known as self-organisation because the network receives no external guidance. Unsupervised learning algorithms enable the network to learn something about the statistical properties of the input patterns with neural units often behaving like feature detectors. There are two classes of unsupervised learning: (a) *coincidence learning* where the weights are modified in response to events that occur simultaneously and (b) *competitive learning* where neural units compete for the privilege of learning.

4.5 Neural network capabilities

Imaginative research continuously finds new uses for artificial neural networks. Some of the more traditional applications include:

Classification - Neural networks, for example, can be used to determine crop types from satellite photographs, to distinguish a submarine from a boulder given its sonar return, and to identify diseases of the heart from electrocardiograms. Any task that can be done by traditional discriminant analysis can be done at least as well, and in a significant number of cases better, by a neural network.

Noise reduction - An artificial neural network can be trained to recognise a number of patterns. These patterns may be parts of time-series, images, etc. If a version of one these patterns, corrupted by noise, is presented to a properly trained network, the network can provide the original pattern on which it was trained. This technique has been used with great success in some image restoration problems.

Prediction - An often-encountered problem is that of predicting the value of a variable given historic values of itself (and perhaps other variables). Economic and meteorological models are but examples. Neural networks have frequently been shown to outperform traditional techniques like ARIMA and frequency domain analysis.

Haykin (1994) states that artificial neural networks are most likely to be superior to other methods under the following conditions:

1. The data on which the conclusions are to be based is "fuzzy". If the input data is human opinions, ill-defined categories, or is subject to possibly large errors, the robust behaviour of neural networks is important.
2. The patterns important to the required decision are subtle or deeply hidden. One of the principal advantages of a neural network is its ability to discover patterns in data that are so obscure as to be imperceptible to human researchers and standard statistical methods. One of the first major commercial uses of neural networks was predicting the credit-worthiness of loan applicants based on their spending and payment history. The correct decision depends on far more than simple factors like salary and debt level. Neural networks were shown to provide decisions superior to those made by trained humans.
3. The data exhibits significant unpredictable nonlinearity. Traditional time series models for predicting future values, such as ARIMA and Kalman filters, are based on strictly defined models. If the data does not fit the models, results will be useless. Neural networks are marvellously adaptable.
4. The data is chaotic (in the mathematical sense). Chaos can be found in telephone line noise, stock market prices, and a host of physical processes. Such behaviour is devastating to most other techniques, but neural networks are generally robust with the inputs of this type.

The good performance of neural networks is not surprising when one considers the solid theoretical foundations on which many of them rest. The standard three-layer feedforward network has powerful function-approximation capabilities. In particular, any continuous function defined over a compact subset of the real number domain can be approximated to an arbitrary accuracy given sufficient hidden neurons. This result is important; when combined with the robustness of the three-layer feedforward network as regards input errors, it is a powerful tool. Rigorous and mathematical discussion of these properties is given in Hornik (1991), and Blum et al.(1991).

In summary, many artificial neural networks possess both substantial theoretical foundations and practical utility. Any problem that can be solved with traditional modelling or statistical methods can most likely be solved more effectively with a neural network.

4.6 The types of networks

There are many different types of neural networks. Characteristic of each type is a highly parallel processing capability arising from an interconnected network of simple computational elements. The neural networks differ from each other in architecture and training algorithm. The *Hopfield* neural network is important for historical reasons. John Hopfields pioneering work gave credibility to the neural network field in the early 1980s. The networks bearing his name are useful for image recall. Partial images can be input to the network and a full image will be produced as an output. These neural networks also have interesting properties as dynamical systems. The *multi-layer feedforward*, or so-called *backpropagation*, network is responsible for most of the successful applications of neural networks and is by far the most commonly used neural network. The *Kohonen self-organising* network, also referred to as SOM – Self Organising Map, is significant as an example of a network capable of unsupervised learning. That is, this neural network does not have to be supplied with a 'correct' answer for each input. It has the capability of sorting the input data into categories. This type of network can also be applied to practical control problems (Welstead, 1994). In 1988, Broomhead et al. described a procedure for the design of layered feedforward networks using *radial basis functions* (RBF), which provide an alternative for multi-layer perceptrons. This linked the design of neural networks to an important area in numerical analysis and linear adaptive filters. The *real-time recurrent* neural network (William et al., 1989) differs from a Hopfield network, which is also a recurrent network, in two important aspects: (a) the network contains hidden neurons and (b) it has arbitrary dynamics. Of particular interest is the ability of the recurrent network to deal with time varying input or output through its own temporal operation (see *Chapter* 6).

4.6.1 The Hopfield network

In 1982, John Hopfield published a paper describing a neural network, the collective properties of which produced a *content-addressable memory*. In contrast to conventional computer memories where stored information is accessed by knowing its address, a content-addressable memory retrieves stored data on the basis of partial information. Thus, a content-addressable memory yields an entire memory item from any subpart of sufficient size. In general, a content-addressable memory is characterised by a number of locally stable states to which the system is attracted. The locally stable states correspond to the stored patterns. Thus, from a given initial state $x = \xi_p + \delta$, which represents partial knowledge of the memory ξ_p , the system should converge to the locally stable state ξ_p.

Architecture

This neural network, which has since become known as the Hopfield network, consists of a single layer of fully connected neural units (*Figure 4.5*). Each neural unit receives a single input from the environment and, therefore, the network does not need an input layer of fan-out elements. All inputs are applied simultaneously, after which the network passes through a series of states until it reaches a stable state, which is the network's output. To ensure that the network converges to a stable state, the weights between neural units must be symmetrical:

$$w_{ji} = \begin{cases} w_{ij} & i \neq j \\ 0 & i = j \end{cases} \tag{4.6}$$

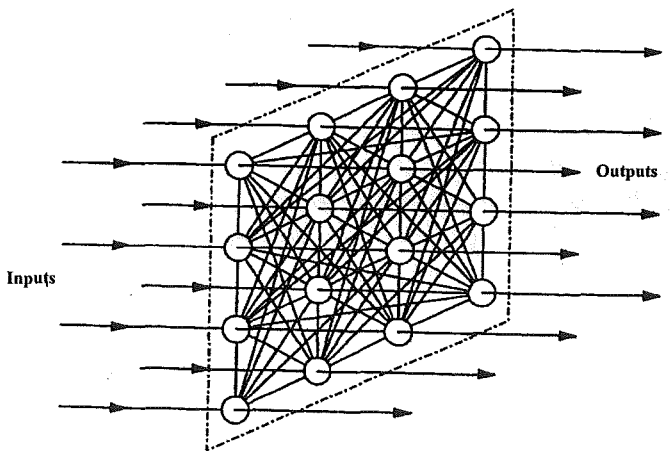


Figure 4.5 – **Architecture of the Hopfield network.**

Neural Units

The neural units of the Hopfield network have two possible output values (0 and 1). These binary neural units can be expressed as:

$$y_j = \mu\left[\sum_{i=1}^{M} w_{ji}\, x_i \; - \; \theta_j\right] \tag{4.7}$$

where $\mu\,[\cdot]$ is the unit step function (*Figure 4.3a*) and the inputs x_i are binary valued. Typically, the threshold θ_j is zero and each neural units randomly updates its output at an average rate R. The asynchronous operation of the neural units represent a combination of the propagation delays, jitter and noise that are present in real neural systems.

Learning Algorithms

For the Hopfield network to operate as a content-addressable memory, the weights between the neural units are set as follows:

$$w_{ji} = \begin{cases} \frac{1}{M}\sum_{p=1}^{P} \xi_{pi}\, \xi_{pj} & i \neq j \\ 0 & i = j \end{cases} \tag{4.8}$$

where M is the number of neural units, P is the number of patterns to be stored and ξ_{pi} is the i^{th} element of the pattern ξ_p. The weights of the network are calculated after a single pass through the training patterns.

Properties

The collective properties of the Hopfield network produce a content-addressable memory for binary patterns. Hopfield analysed the performance of the neural network as a content-addressable memory by considering its energy function E :

$$E = -\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} w_{ji}\, y_i y_j \tag{4.9}$$

The energy function, also known as the *energy landscape*, is characterised by a number of hollows or *local minima*, which corresponds to the stored patterns. When the output of a neural unit changes, the energy of the system decreases and, thus, the state of the network

converges to a local minimum. Unfortunately, the Hopfield network does not always converge to the nearest local minimum and, consequently, does not always yield the correct memory. A further problem arises if one attempts to store too many patterns in the network. In this case, spurious local minima are generated that do not correspond to any of the training patterns.

Alternative Implementations

Several variations of the Hopfield network exist, the most common of which is to use bipolar neural units whose two possible outputs are ±1 (Lippmann, 1987). The inputs to this network are also bipolar which allows the weights between neural units to become negative. Hopfield (1984) showed that the network still behaved as a content-addressable memory even if the neural units had a graded response (e.g., a sigmoidal transfer function) and were updated synchronously.

When stored patterns are not orthogonal, interference between patterns occurs and local minima are produced in the energy landscape which do not correspond to stored patterns. Often these local minima are not as 'deep' as those corresponding to stored patterns. Therefore, the network needs a way to escape from shallow local minima. Addition of thermal noise allows the network to move to higher states of energy and escape from local minima. This is usually implemented through a probabilistic update rule, where the probability that the next output of a neural unit is 1 is given by:

$$p_j = \frac{1}{1+e^{-\Delta E_j T}} \tag{4.10}$$

$$\Delta E_j = \sum_{i=1}^{M} w_{ji} x_i - \theta_j$$

where T is a parameter that acts like the temperature of a physical system. The temperature T decreases with time and, therefore, this process is known as *simulated annealing* (Ackley et al., 1985; Kirkpatrick et al., 1983).

4.6.2 The Self-organising map

In 1982, Teuvo Kohonen introduced a neural network that is able to discover important features of the training patterns. Furthermore, this network, which is known as the self-organising map, forms spatially ordered representations of these features so that the location of the active neural unit is specific to a certain characteristic feature of the input pattern. For example, the location of a neural unit may correspond to the frequency of the input signal.

Architecture

The self-organising map consists of a single planar array of neural units (*Figure 4.6*). Each neural unit receives the entire input pattern through a set of weighted connections.

Neural Units

the neural units which make up the self-organising map are linear and can be expressed as :

$$y_j = \sum_{i=1}^{M} w_{ji} x_{pi} \tag{4.11}$$

This neural unit can also be expressed in vector notation as:

$$y_j = \mathbf{w}_j \bullet \mathbf{x}_p \tag{4.12}$$

which is the scalar or dot product of the input pattern $\mathbf{x}_p$ with the weight vector $\mathbf{w}_j$ of neural units j. The dot product is a measure of their similarity. Therefore, a linear neural unit

compares the input pattern with its weight vector and responds with a measure of the degree of matching. Typically, all vectors are normalised to unit length (i.e., $\| \mathbf{w}_j \| = 1$ and $\| \mathbf{x}_p \| = 1$) and, thus, the output of the neural unit becomes the cosine of the angle θ_{pj} between the vectors :

$$\begin{aligned} y_j &= \mathbf{w}_j \bullet \mathbf{x}_p \\ &= \| \mathbf{w}_j \| \ \| \mathbf{x}_p \| \cos \theta_{pj} \\ &= \cos \theta_{pj} \end{aligned} \tag{4.13}$$

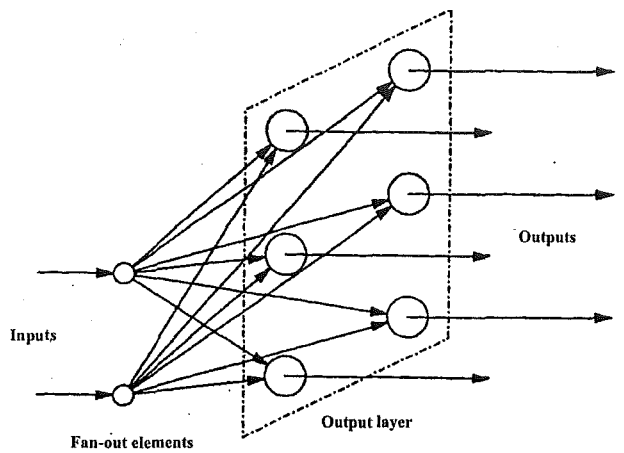


Figure 4.6 – **Architecture of the self-organising map.**

Learning Algorithm

The self-organising map employs an unsupervised competitive learning algorithm where neural units compete for the privilege to learn. The winning unit j^* is the one with the largest output value:

$$\begin{aligned} y_{j^*} &= \max_j \ (y_j) \\ &= \max_j \ (\mathbf{w}_j \bullet \mathbf{x}_p) \\ &= \max_j \ (\cos \theta_{pj}) \end{aligned} \tag{4.14}$$

Because the vectors have been normalised to unit length, the maximum output corresponds to the minimum angle θ_{pj} between the input pattern and the weight vector.

The weight vectors of the winning unit and its neighbours are modified according to the rule:

$$\mathbf{w}_j = \frac{\mathbf{w}_j^{old} + \eta \ \mathbf{x}_p}{\left\| \mathbf{w}_j^{old} + \eta \ \mathbf{x}_p \right\|} \tag{4.15}$$

where η is the learning rate and is typically a decreasing function of time. This rule ensures that the weight vectors $\mathbf{w}_j$ remain normalised to unit length and correspond to a rotation around the unit circle towards the input pattern. The weight vectors of all neural units in the *neighbourhood* of the winning unit are adjusted and, thus, the neural units do not learn independently of each other. This interaction of neural units during learning is crucial to the

formation of globally ordered feature maps. In typical implementations of the self-organising map, the size of the neighbourhood around the winning neural unit is initially very large and is slowly decreased with time until it finally includes only the winning unit.

Properties

The competitive learning algorithm employed by the self-organising map causes the neural units to develop into a set of feature-sensitive detectors. Each weight vector moves to the average of the cluster of input patterns for which the corresponding neural unit won the competition to learn. Therefore, the self-organising map clusters the input patterns. Furthermore, the set of weight vectors tends to approximate the probability density function of the input patterns and, thus, the self-organising map also performs vector quantisation (Kohonen, 1990).

If only the weight vector of the winning unit is modified, all neural units operate independently and the order in which they are assigned to clusters is effectively random. However, Kohonen emphasised the importance of the global organisation or ordering of feature maps. The fundamental principle of a topographically organised system is that nearby units must respond similarly and, thus, it is crucial to the formation of ordered maps that neural units do not learn independently. Because the weight vectors of all neural units in the neighbourhood of the winning unit are modified, changes in the individual neural units are only reinforced if they result in global order. As the learning rate and size of the neighbourhood decrease, the mapping moves from being very coarse to being finely tuned which corresponds to global ordering followed by increased selectivity of individual elements.

The self-organising map clusters input patterns on the basis of their characteristic features and displays the overall similarity relations of the input data in a small number of dimensions, limited by our ability to perceive multi-dimensional data. For example, the self-organising map is able to produce mappings that transform a signal pattern of arbitrary dimensionality onto a planar array. Thus, the self-organising map can preserve the topological relations while performing a dimensionality reduction of the input space.

The self-organising map models the statistical properties of the input patterns. However, the model is only as accurate as the size of the network allows. The more neural units available the less area of input space each weight vector must cover and the more accurate the model. For a perfect model there would be one neural unit for each pattern, but this merely states that the ideal model of the input data set is the input data set itself. Therefore, for a network to be capable of discovering anything non-trivial about the data there must be fewer neural units than input patterns.

Alternative implementations

Kohonen (1988) proposed using the Euclidean distance to measure the similarity between a neural unit's weight vector and the input pattern. This eliminates the need to normalise the vectors to unit length. A neural unit of this type can be expressed as :

$$y_j = \left(\sum_{i=1}^{M} (x_{pi} - w_{ji})^2 \right)^{\frac{1}{2}}$$

$$= \left\| \mathbf{x}_p - \mathbf{w}_j \right\| \qquad (4.16)$$

The winning unit j^* is then the neural unit with the minimum output value, which corresponds to the minimum distance between the input pattern and the weight vector:

$$y_{j^*} = \min_j (y_j)$$

$$= \min_j \|\mathbf{x}_p - \mathbf{w}_j\| \tag{4.17}$$

The similarity measure employed must be compatible with the learning law (Kohonen, 1988) and, therefore, the rule by which the winning neural unit and its neighbours modify their weight vector becomes:

$$\mathbf{w}_j = \mathbf{w}_j^{old} + \eta \left(\mathbf{x}_p - \mathbf{w}_j^{old}\right) \tag{4.18}$$

The choice of metric (i.e. scalar product or Euclidean distance) determines how the input patterns are clustered. In fact, the Euclidean distance version is effectively a neural network implementation of the k-means clustering procedure (Hunt, 1975), which also spatially orders the clusters.

4.6.3 The Multi-layer Feedforward (Back-propagation) network

Multi-layer Feedforward neural networks are effectively multi-layer Perceptrons that employ the supervised learning algorithm known as *back-propagation*. These so-called back-propagation networks are applicable to a wide variety of problems because they are able to learn arbitrary associations between input and output patterns. Consequently, these networks are also the most common.

Architecture

A typical back-propagation neural network is made up of several layers with feedforward connections between neural units of consecutive layers (*Figure 4.1*). Each neural unit receives inputs from all neural units of the previous layer and projects its output to all those of the following layer.

Multi-layer networks are able to solve very complex problems. Whereas single layer networks can only distinguish classes that are linearly separable (Minsky et al, 1969), three layer networks can discriminate arbitrarily complex classes (Beale et al., 1990; Lippmann, 1987).

Neural Units

Multi-layer networks of linear units can be reduced to single layer networks that can only distinguish linearly separable classes (Minsky et al., 1969). Therefore, the neural units of a back-propagation network are non-linear and have the following form:

$$y_j = f\left[net_j\right]$$

$$net_j = \sum_{i=0}^{M} w_{ji}\, x_i \tag{4.19}$$

where $f[\cdot]$ is some non-linear, non-decreasing function. The back-propagation learning algorithm further requires that $f[\cdot]$ is differentiable and, typically, a sigmoid (*Figure 4.4a*) is employed:

$$f[net] = \frac{1}{1 + e^{-net}} \tag{4.20}$$

Learning Algorithm

The back-propagation learning algorithm (Rumelhart et al., 1986) is a supervised learning algorithm that modifies the weights w_{ji} by an amount Δw_{ji} in order to minimise the mean squared error function E. The mean squared error for a given training pattern $\mathbf{x}_p$ is :

$$E_p = \frac{1}{2}\sum_j \left(t_{pj} - y_{pj}\right)^2 \tag{4.21}$$

where t_{pj} is the target output of a neural unit j for pattern p and y_{pj} is its actual response. The mean squared error for the entire training set of P patterns is the summation of the error for each pattern:

$$E = \sum_{p=1}^{P} E_p \tag{4.22}$$

The back-propagation algorithm minimises the error function E by gradient descent, adjusting the weights by an amount proportional to $-\partial E / \partial w_{ji}$:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \tag{4.23}$$

where η is the learning rate ($\eta > 0$). The partial derivative $\partial E / \partial w_{ji}$ can be evaluated using the chain rule (Kreyszig, 1983) and the equation for a neural unit as follows:

$$\begin{aligned}\frac{\partial E}{\partial w_{ji}} &= \sum_{p=1}^{P} \frac{\partial E_p}{\partial w_{ji}} \\ &= \sum_{p=1}^{P} \frac{\partial E_p}{\partial y_{pj}} \frac{\partial y_{pj}}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}} \\ &= \sum_{p=1}^{P} - \left(t_{pj} - y_{pj}\right) f'\left[net_{pj}\right] x_{pi}\end{aligned} \tag{4.24}$$

where $f'[x]$ is the derivative df / dx. If $f[\cdot]$ is a sigmoid then its derivative is:

$$f'\left[net_{pj}\right] = y_{pj}\left(1 - y_{pj}\right) \tag{4.25}$$

The weights are adjusted according to the rule:

$$\Delta w_{ji} = \eta \sum_{p=1}^{P} \left(t_{pj} - y_{pj}\right) f'\left[net_{pj}\right] x_{pi} \tag{4.26}$$

Defining a new variable $\delta_{pj} = \left(t_{pj} - y_{pj}\right) f'\left[net_{pj}\right]$ allows the weight change to be expressed as:

$$\Delta w_{ji} = \eta \sum_{p=1}^{P} \delta_{pj} x_{pi} \tag{4.27}$$

However, in a multi-layer network the target outputs for neural units in the hidden layer are not known and the partial derivative $\partial E_p / \partial y_{pj}$ must be determined in terms of the units of the output layer. For example, $\partial E_p / \partial y_{pj}$ for units of the last hidden layer can be evaluated as follows:

$$\frac{\partial E_p}{\partial y_{pj}} = \sum_k \frac{\partial E_p}{\partial y_{pk}} \frac{\partial y_{pk}}{\partial y_{pj}}$$

$$= \sum_k (t_{pk} - y_{pk}) f'[net_{pk}] w_{kj}$$

$$= \sum_k \delta_{pk} w_{kj} \tag{4.28}$$

where the summation with respect to k is over all neural units in the output layer. The error between the target pattern and the actual output is effectively propagated backwards through the network to the neural units of the previous layer. In fact, the result of (4.28) holds for neural units of all hidden layers and, thus, δ_{pj} is given by:

$$\delta_{pj} = \begin{cases} (t_{pj} - y_{pj}) f'[net_{pj}] & \textit{for output layer} \\ f'[net_{pj}] \sum_k \delta_{pk} w_{kj} & \textit{for hidden layers} \end{cases} \tag{4.29}$$

where the summation with respect to k is taken over all units in the following layer. All neural units adjust their weights by an amount Δw_{ji} which is determined by the equation:

$$\Delta w_{ji} = \eta \sum_{p=1}^{P} \delta_{pj} x_{pi} \tag{4.30}$$

Because the weights are only adjusted after an entire pass through the training set, this is known as the *batch* version of the algorithm. However, the weights can be adjusted after presentation of each pattern as follows:

$$\Delta w_{ji} = \eta \delta_{pj} x_{pi} \tag{4.31}$$

This is known as pattern learning and, provided η is sufficiently small, also performs gradient descent. However, when η is small, learning is very slow. By introducing a momentum factor, η can be increased without causing instability. The learning algorithm becomes:

$$\Delta w_{ji} = \eta \delta_{pj} x_{pi} + \alpha \Delta w_{ji}^{old} \tag{4.32}$$

where α is a positive constant ($0 \leq \alpha \leq 1$).

Eberhart et al. (1990) included a momentum term in the batch mode learning algorithm as follows:

$$\Delta w_{ji} = \eta \sum_{p=1}^{P} \delta_{pj} x_{pi} + \alpha \Delta w_{ji}^{old} \tag{4.33}$$

where again α is a positive constant ($0 \leq \alpha \leq 1$). This helps increase the learning speed when the gradient of the weight space is small.

Properties

The back-propagation learning algorithm provides a method for training multi-layer neural networks, which are considerably more powerful than single layer networks. Multi-layer networks are able to learn arbitrary associations between the input and output patterns by forming internal representations of the input patterns. In effect, the hidden layers encode the features of the input patterns that the neural network considers to be important.

Unfortunately the training of back-propagation networks is slow and the network weights may not converge to the global minimum of the error function but may become trapped in a local minimum.

Alternative implementations

The collective computational power of back-propagation networks is relatively insensitive to the details of the neural transfer function $f[\cdot]$. Therefore, functions other than the sigmoid can be used. For example, the *tanh* function (*Figure 4.2c*) is commonly used and Tepedelenlioglu et al. (1989) showed that non-differentiable piecewise linear functions could also be used provided that a suitable derivative function $f'[\cdot]$ is was defined.

A considerable number of modifications to the back-propagation learning algorithm exist in an attempt to decrease training time. For example, the introduction of a momentum term as in (4.32) allows the learning rate to be increased without causing the algorithm to be unstable. Momentum decreases back-propagation's sensitivity to small details in the error surface. This helps the network avoid getting stuck in shallow minima that would prevent the network from finding a lower error solution. Training time can also be decreased by the use of an adaptive learning rate which attempts to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface (Orr et al., 1998).

Hagan (1996) advocates Levenberg-Marquardt optimisation to make training times even shorter; it is roughly in the order of 100 times faster than standard back-prop. Levenberg-Marquardt optimisation is a more sophisticated method than gradient descent (back-propagation). Its major limitation is that it requires a great deal of processing memory for large problems.

Use of *tanh* as the transfer function of neural units tends to decrease the training time, because it does not approach zero at negative infinity. However, at the beginning of the training, when the weights are small and all net inputs close to zero, learning may be slow. When the output of a neural unit is very close to 0 or 1 very little learning takes place because the value of the derivative is almost zero. Tveter (1991) suggested that this problem may be overcome, to some extent, by adding a small positive constant to the derivative. Similarly, Haffner et al. (1989) proposed adding a linear function to the transfer function, which results in a constant being added to the derivative.

The training strategy employed can also improve training speed. For example, it is common for back-propagation networks to be trained first on easy examples before more difficult or borderline examples are presented. This is similar to the strategy proposed by Caudill (1991) where the error is set to zero if it is less than a given error tolerance. Initially the error tolerance is large but is decreased over the training session.

The order in which the training examples are presented to the network should be *randomised* (shuffled) from one epoch to the next. This form of randomisation can be critical for improving the speed of convergence. Also, the use of queries may improve the training efficiency (Baum, 1991).

It is possible to overtrain a back-propagation network so that it begins to memorise the training patterns and, therefore, loses its ability to generalise. Caudill (1991) suggested adding a small amount of noise to the training patterns so that the network never sees exactly the same pattern twice. In this way, generalisation rather than memorisation is encouraged.

4.6.4 The Radial basis network

The back-propagation algorithm for the design of a multi-layer perceptron (under supervision) as described in section 4.6.3 may be viewed as an application of an optimisation method known in statistics as *stochastic approximation*. A different approach is to view the design of a neural network as a *curve-fitting (approximation) problem* in a high-dimensional space. Accordingly, learning is equivalent to finding a surface in a multi-dimensional space that provides a best fit to the training data. Any generalisation is equivalent to the use of this multi-dimensional surface to interpolate the test data.

This is approach is the motivation behind the method of radial basis functions. In the context of a neural network, the hidden units provide a set of "functions" that constitute an arbitrary "basis" for the input patterns (vectors) when they are expanded into the hidden unit space; these functions are called *radial basis functions* (*Figure 4.7*).

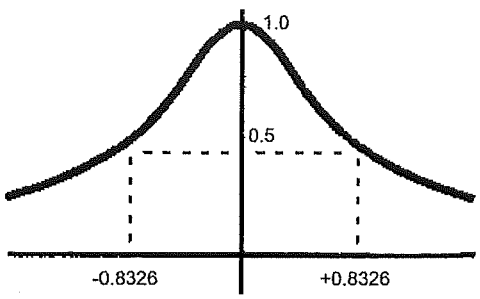


***Figure 4.7* – The radial basis function.**

Radial Basis networks usually require considerably more neurons than standard feed-forward backpropagation networks, the reason being that neurons with a sigmoid function can have outputs over a large region of the input space, while radial basis neurons only respond to relatively small regions of input space. The result is that the larger the input space (in terms of the number of inputs, and the ranges those inputs vary over) the more radial basis neurons are required. But often they can be designed in a fraction of the time it takes to train standard feed-forward networks. They only work in an optimal manner when a large set of training data is available.

Architecture

The radial basis function (RBF) network in its most basic form involves three entirely different layers (*Figure 4.8*). The input layer is made up out of source nodes (sensory units). The second layer is a hidden layer of high enough dimension, which serves a different purpose from that in a multi-layer perceptron. The output layer supplies the response of the network to the activation patterns applied to the input layer.

Neural Units

The number of units in the hidden layer of the generalised RBF network is ordinarily smaller than their number of examples available for training. The linear weights associated with the output layer, and the positions of the centres of the radial-basis functions are all unknown parameters that have to be learned. Provision is made for a bias (i.e. data-independent variable) applied to the output unit simply by setting one of the linear weights in the output layer of the network equal to +1.

A radial basis neuron receives as net input the vector distance between its weight vector **w** and the input vector **p**, multiplied by the bias **b**. As the distance between **w** and **p** decreases, the output increases. Thus a radial basis neuron acts as a detector which outputs a value of 1 whenever the input is identical to its weight vector **w**.

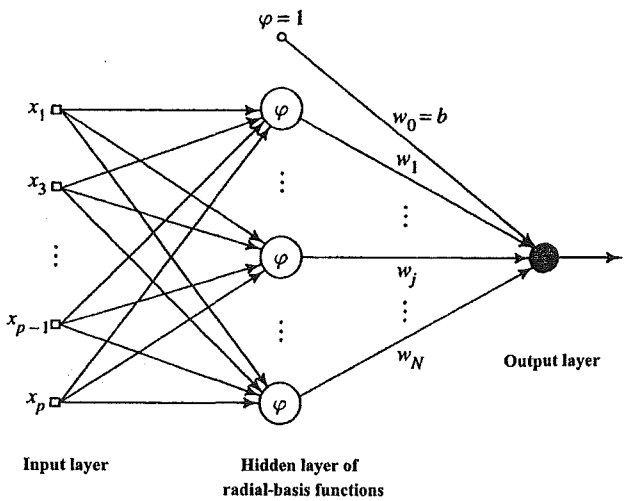


Figure 4.8 – **An RBF network with non-linear radial basis functions in the hidden layer and a linear function in the output layer.**

The transformation from the input space to the hidden unit space is *non-linear*, whereas the transformation from the hidden unit space to the output space is *linear*. A pattern-classification problem cast in a high-dimensional space non-linearly is more likely to be linearly separable than in a low-dimensional space - hence the reason for making the dimension of the hidden unit space in a RBF network high (Haykin, 1994). Through careful design, however, it is possible to reduce the dimension of the hidden unit space, especially if the centres of the hidden units are made adaptive.

Training

The learning process undertaken by a radial–basis function network may be visualised on two levels. The linear weights associated with the output units of the network tend to evolve on a different time-scale compared to the non-linear activation functions of the hidden units. Thus, as the hidden layers activation functions evolve slowly in accordance with some non-linear optimisation strategy, the output layers waits adjust themselves rapidly through a linear optimisation strategy. The different layers of an RBF network perform different tasks, and so it is reasonable to separate the optimisation of the hidden and output layers of the network by using different techniques, and perhaps operating on different time scales (Lowe, 1991).

One approach is to assume fixed radial–basis functions defining the activation functions of the hidden units. The functions can be Gaussian whose centres may be chosen randomly from the training data set. The only parameters that would be to the learned in this approach are the linear weights in the output layer of the network. A procedure for doing this is to use the *pseudo inverse* method as given in Broomhead et al. (1988).

A second approach is to have the centres of the radial–basis functions and all other free parameters of the network undergo a supervised learning process; in this way the network takes on its most generalised form. The error correction meaning is implemented using a gradient descent procedure that represents a generalisation of the LMS algorithm.

Wettscherek et al. (1992) have compared the performance of the two above mentioned techniques with that of the multilayer perceptron. They found that RBF networks with *unsupervised* learning of the centres locations and *supervised* learning of the output layer

weights did not generalise nearly as well as multi–layer perceptrons trained with the back–propagation algorithm. However, generalised RBF networks with *supervised* learning of the centres locations as well as the output layer weights were able to exceed substantially the generalisation performance of multi–layer perceptrons.

Properties

Radial basis functions were first introduced in the solutions of the real multi-variate interpolation problem. Powell (1985) surveys the early work on this subject. Broomhead et al (1988) were the first to exploit the use of radial basis functions in the design of neural networks. Other major contributions to the theory, design, and application of radial basis function networks include papers by Moody et al (1989), Renals (1989), and Poggio et al (1990a). This last paper emphasises the use of the *regularisation* theory, applied to this class of neural networks, as a method for improved generalisation to new data.

The RBF network is designed to perform a non-linear mapping from the input space to the hidden space and a linear mapping from the hidden to output space. Then, in an overall fashion, the network represents a map from the p-dimensional input space to the d-dimensional output space, written as:

$$s: \mathrm{R}^{\mathrm{p}} \Rightarrow \mathrm{R}^{\mathrm{d}} \tag{4.34}$$

The map s is a *hypersurface* (graph) $\Gamma \subset \mathrm{R}^{\mathrm{p}+1}$, where the surface Γ is a multi-dimensional plot of the output as a function of the input. In practice, the surface Γ is unknown and the training data is usually contaminated with noise. Accordingly, the training phase and the generalisation phase of the learning process may be respectively viewed as follows (Broomhead et al, 1988):

1. The training phase constitutes the optimisation of a fitting procedure for the surface Γ, based on known data points presented to the network in the form of input-output patterns.
2. The generalisation phase is synonymous with interpolation between the data points, with the interpolation being performed along the constrained surface generated by the fitting procedure as the optimum approximation to the true surface Γ.

The interpolation problem may be stated as follows (making use of a single output),

Given a set of N different points $\{ \mathbf{x}_i \in \mathrm{R}^{\mathrm{p}} \mid i = 1, 2, ..., N \}$ and a corresponding set of N real numbers $\{ d_i \in \mathrm{R}^1 \mid i = 1, 2, ..., N \}$, find a function $F: \mathrm{R}^{\mathrm{p}} \Rightarrow \mathrm{R}^1$ that satisfies the interpolation condition:

$$\mathrm{F}(\mathbf{x}_i) = d_i, \quad i = 1, 2, ..., N \tag{4.35}$$

Note that for the interpolation specified here, the interpolating surface (i.e., function F) is constrained to pass through *all* the training points.

The *radial basis functions* (RBF) technique consists of choosing a function F that has the following form (Powell, 1988);

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi \left(\| \mathbf{x} - \mathbf{x}_i \| \right) \tag{4.36}$$

where $\left\{ \varphi \left(\| \mathbf{x} - \mathbf{x}_i \| \right) \mid i = 1, 2, .., N \right\}$ is a set of N arbitrary (generally non-linear) functions, known as radial basis functions, and $\| \bullet \|$ denotes a *norm* that is usually taken to be Euclidean. The known data points $\mathbf{x}_i \in \mathrm{R}^{\mathrm{p}}$, $i = 1, 2, ..., N$ are taken to be the *centres* of the radial basis functions. An often used function for the non-linearity φ is the Gaussian function;

$$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \qquad \text{for } \sigma > 0, \text{ and } r \geq 0 \tag{4.37}$$

which allows for the construction of an N x N interpolation matrix ϕ which is positive definite (Light's theorem) according to Light (1992). Provided that the data points are all distinct it is possible to obtain the N x 1 linear weight vector $\mathbf{w}$ by utilising the inverse of the interpolation matrix.

$$\mathbf{w} = \phi^{-1}\,\mathbf{d} \tag{4.38}$$

There are a number of disadvantages with the solution to the (strict) interpolation problem. In theory equation 4.38 can always be solved; in practice there is no solution when the matrix ϕ is arbitrarily close to singular. It is also not a good strategy for training RBF networks for certain classes of tasks because of poor generalisation to new data. This lack of generalisation is for the following reason; when the number of data points in the training set is much larger than the number of degrees of freedom of the underlying physical process, and the constraint is to have as many radial basis functions as data points, the problem is overdetermined. Consequently, the network may end up fitting misleading variations due to idiosyncrasies or noise in the input data, thereby resulting in a degraded generalisation performance (Broomhead et al, 1988).

It was previously noted that *learning* could be viewed as a hypersurface reconstruction problem, given a set of data points that may be sparse. According to this viewpoint, the hypersurface reconstruction or approximation problem belongs to a generic class of problems known as *inverse problems.* An inverse problem may be *well-posed* or *ill-posed* (Morozov, 1993).

Learning is an ill-posed inverse problem for the following reasons. First, there is not as much information in the training data as is needed to reconstruct the input-output mapping uniquely. Second, the presence of noise or imprecision in the input data adds uncertainty to the reconstructed input-output mapping. To make the learning well-posed so that generalisation to new data is feasible, some from of prior information about the input-output mapping is needed (Poggio et al, 1990a). This, in turn, means that the process responsible for the generation of the input-output examples used to train a neural network must exhibit *redundancy* in information-theoretic sense. Most practical physical processes (e.g., speech, pictures, radar signals, sonar signals, seismic data, etc.) satisfy this requirement.

The *regularisation theory* offers a solution for ill-posed problems (Morozov, 1993). In the context of approximation problems, the basic idea of regularisation is to stabilise the solution by means of some auxiliary non-negative functional that embeds prior information, and thereby make an ill-posed problem into well-posed one (Poggio, 1990a).

Let the approximating function be denoted by $F(\mathbf{x})$; a solution to the regularisation problem is given by the expansion

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i G\left(\mathbf{x};\mathbf{x}_i\right) \tag{4.39}$$

where $G(\mathbf{x};\mathbf{x}_i)$ is the *Green's function*, centred at $\mathbf{x}_i$, for the *self-adjoint differential operator* $\mathbf{P^*P}$ (Poggio, 1990a), (taking adjoints is similar to the conjugation of complex numbers), and w_i is the ith element of the weight vector $\mathbf{w}$.

The minimising solution $F(\mathbf{x})$ to the regularisation problem is a linear superposition of N Green's functions (Girosi et al, 1990a). The Green's function plays the same role for a linear

differential equation as does the inverse matrix for a matrix equation. Equation 4.39 states the following:

- The regularisation approach is equivalent to the expansion of the solution in terms of a set of Green's functions, whose characterisation depends only on the form adopted for the stabiliser **P** and the associated boundary conditions.
- The number of Green's functions used in the expansion is equal to the number of examples used in the training process.

The characterisation of the Green's function, for a specified centre $\mathbf{x}_i$, depends only on the form of the stabiliser **P**, that is, on the *a priori* assumption made concerning the input-output mapping. If **P** is both *translationally and rotationally invariant*, then the Green's function will depend only on the *Euclidean norm* of the difference vector $\mathbf{x} - \mathbf{x}_i$; that is

$$G(\mathbf{x};\mathbf{x}_i) = G\left(\|\mathbf{x} - \mathbf{x}_i\|\right) \tag{4.40}$$

Under these conditions, the Green's function must be a radial basis function. In such a case, the regularised solution of equation 4.39 takes on the following special form (Poggio, 1990a):

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i G\left(\|\mathbf{x} - \mathbf{x}_i\|\right) \tag{4.41}$$

This interpolating function $F(\mathbf{x})$ constructs a linear function space and is generated by all known training data points N.

Alternative implementations

The expansion of the approximating function $F(\mathbf{x})$ given in equation 4.39 in terms of the Green function $G(\mathbf{x};\mathbf{x}_i)$ centred at $\mathbf{x}_i$ offers the network structure shown in *Figure 4.8* as a method for its implementation.

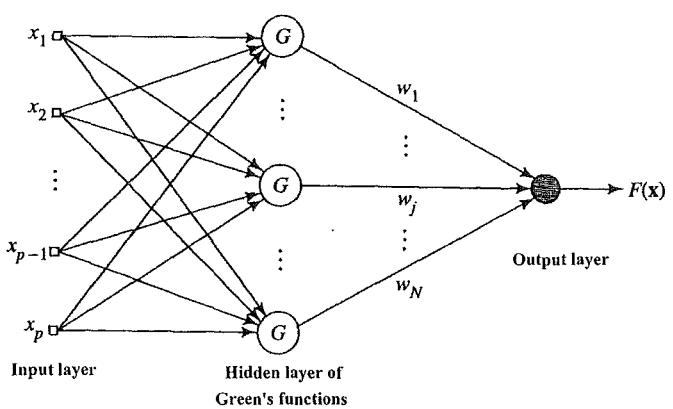


***Figure 4.8* – An RBF regularisation network with Green's functions being utilised in the single hidden layer.**

This, known as the *regularisation network*, consists of three layers. The hidden layer has a number of hidden units *equivalent* to the number of training samples. The activation functions of the individual hidden units are defined by the Green's functions. The output layer consists of a single linear unit.

From the viewpoint of *approximation theory* the regularisation network has the desirable property of being a *universal approximator*. Meaning that it can approximate arbitrarily well

any multi-variate continues function given a sufficiently large number of hidden units. This is also its weakness; having to utilise all N data points produces a matrix that is prohibitively expensive to implement in computational terms for large N.

4.7 Hardware implementations

One of the major advantages of neural networks is their speed due to the massive parallel architecture. Because of the difficulty implementing neural networks in hardware, computer simulations are often performed. However, such simulations are slower because all computations are carried out sequentially. For certain complex applications such as speech recognition and optical character recognition, a high level of performance is required; a performance that would be better served by dedicated hardware.

Hardware implementation of neural networks is an active area of research. The major difficulty facing the researchers is the huge number of connections required between neural units. In recent years many computing hardware ideas have been explored including electronic, optical, mechanical, acoustic and chemical implementations (Hecht-Nielsen, 1990). One area that receives a great deal of attention is VLSI implementation, a technology that provides an ideal medium for the hardware implementation of neural networks. VLSI technology allows integrated circuits with tens of millions of transistors on a single silicon chip, and this number can be increased by two orders of magnitude before reaching the fundamental limits of the technology imposed by the law of physics (Hoeseinen et al, 1972; Keyes, 1987). Boser et al. (1992) state that VLSI technology is well matched to neural networks for two principal reasons:

- The high functional density achievable with VLSI technology permits the implementation of a large number of identical, concurrently operating neurons on a single chip, thereby making it possible to exploit the inherent parallelism of neural networks.
- The regular topology of neural networks and the relatively small number of well-defined arithmetic operations involved in their learning algorithms greatly simplify the design and layout of VLSI circuits.
- At present there are general-purpose chips available for the construction of multi-layer perceptrons, Boltzmann machines, mean-field theory machines, and self-organising neural networks. In addition, special-purpose chips have been developed for specific information-processing functions (Haykin, 1994).

4.8 Comparing Neural Networks with Statistical methods

There is considerable overlap between the fields of neural networks and statistics. Statistics is concerned with data analysis. In neural network terminology, statistical inference means learning to generalise from noisy data. Some neural networks are not concerned with data analysis (e.g., those intended to model biological systems) and therefore have little to do with statistics. Some neural networks do not learn (e.g., Hopfield nets) and therefore have little to do with statistics. Some neural networks can learn successfully only from noise-free data (e.g., ART or the perceptron rule) and therefore would not be considered statistical methods. But most neural networks that can learn to generalise effectively from noisy data are similar or identical to statistical methods (Sarle, 1994). For example:

Feedforward nets with *no* hidden layer (including functional-link neural nets and higher-order neural nets) are basically *generalised linear* models.

Feedforward nets with *one hidden* layer are closely related to *projection pursuit regression.*

Probabilistic neural nets are identical to kernel *discriminant analysis*.

Kohonen nets for adaptive vector quantisation are very similar *to k-means cluster analysis*.

Hebbian learning is closely related to *principal component analysis*.

A neural network that appears to have no close relative in the existing statistical literature is Kohonen's self-organising maps (SOM).

Feedforward nets are a subset of the class of *nonlinear regression and discrimination models*. Statisticians have studied the properties of this general class but had not considered the specific case of feedforward neural nets before such networks were popularised in the neural network field (Sarle, 1997). Nevertheless, many results from the statistical theory of non-linear models apply directly to feedforward nets, and the methods that are commonly used for fitting non-linear models, such as the *Levenberg-Marquardt* and *conjugate gradient* algorithms, can be used to train feedforward nets.

While neural nets are often defined in terms of their algorithms or implementations, statistical methods are usually defined in terms of their results. The arithmetic mean, for example, can be computed by a (very simple) back-propagation network. The end-result is still an *arithmetic mean* regardless of the way it is computed. A statistician would thus consider standard back-propagation and Levenberg-Marquardt as different algorithms for implementing the same statistical model such as a feedforward network. On the other hand, different training criteria, such as least squares and cross entropy, are viewed by statisticians as fundamentally different estimation methods with different statistical properties. Bishop (1995) and Ripley (1996) explore the application of statistical theory to neural networks in detail.

Communication between statisticians and neural net researchers is often hindered by the different terminology used in the two fields. There is a comparison of neural net and statistical jargon in ftp://ftp.sas.com/pub/neural/jargon.

4.9 Summary

Much of the current research effort on neural networks focuses on pattern classification. Given the practical importance of pattern classification and its rather pervasive nature, and the fact that neural networks are so well suited for the task of pattern classification, this concentration of research effort is understandable. However, for artificial neural networks to be used with the envisaged EMS it is necessary to start thinking of pattern classification in a much broader sense; one where *adaptability* and combination with a *real world* system leads to solving classification, control and prediction problems of a more complex and sophisticated nature.

Control, an area of application naturally suited for neural networks, is also evolving in its own way in the direction of *intelligent control*. This ultimate form of control is defined as the as the ability of a system to comprehend, reason, and learn about processes, disturbances and operating conditions (Åström et al, 1992). As with intelligent pattern classification, the key attribute that distinguishes intelligent control from classical control is the extraction and exploitation of *knowledge* for improved system performance. The fundamental goals of intelligent control may be described as follows (White et al.,1992):

1. Full utilisation of knowledge of a system and/or feedback from a system to provide *reliable control* in accordance with some pre-assigned performance criterion.

2. Use of the knowledge to control the system in an *intelligent manner*, as a human expert may function in light of the same knowledge.
3. Improved ability to control the system over time through the accumulation of experiential knowledge (i.e. learning from experience).

This list of goals is ambitious but sums up nicely what is trying to be achieved with the energy management system. However, it is the author's opinion that it cannot be attained at this stage by the use of neural networks alone. Rather, it is necessary to resort to the combined use of neural networks with some sort of dynamic programming.

The next chapter examines the prominence of various forms of artificial neural networks in real-world problem solving and serves to demonstrate their versatility.

Chapter 5. Applications of Neural Networks

5.1 Introduction

Because of their ability to learn from examples, neural networks can be applied to a wide range of tasks and are particularly applicable to those problems for which mathematical algorithms do not exist or are unsatisfactory. Neural network applications are many and varied; examples include identification of radar patterns, mortgage risk assessment, DNA sequencing, automatic vehicle guidance, content addressable memory, recognition of hand writing, monitoring respiratory function during surgery, image restoration, circuit board layout, adaptive equalisation, parts inspection in manufacturing, intelligent spreadsheets for financial analysts, detection of explosive substances at airports, classification and interpretation of mass spectroscope data, fault diagnosis of engines, speech recognition, target identification in sonar images, image compression and recognition of star patterns for on-board satellite navigation. Despite the diversity of application domains, these tasks can be divided into a number of categories: memory, optimisation, categorisation, control, data processing, predictive and pattern recognition. This Chapter discusses application examples from a number of these areas, detailing network architecture, training and performance as well as problem representation and input data preparation, and concludes with a brief discussion on the limitations of neural networks. With *Chapters* 4 and 5 completed, the reader is considered to have a good general understanding of the various neural network models and their areas of application. It is the preparation for *Chapter* 6, which concentrates on time series prediction using neural networks.

5.2 General considerations

Neural networks are able to perform some tasks that are difficult to implement successfully on conventional digital serial processors. However, neural networks perform poorly on arithmetic tasks at which serial computers excel. Therefore, neural networks must be considered as complementary to, not replacements for, traditional computers and should only be applied to those problems for which conventional computing paradigms are inadequate. Such problems are typically characterised by fuzzy, imprecise or imperfect knowledge or data.

There are a wide variety of neural networks available, only four of which were considered in *Chapter* 4. Each type of neural network has its own functional capabilities and, consequently, one type of neural network cannot perform all tasks. Thus, choice of a neural network depends on the application. For example, if a network is required to discover features in a set of input patterns, then a network employing an unsupervised learning algorithm should be chosen, whereas if the network was to classify input patterns into known categories then a supervised learning algorithm should be employed.

Having chosen a neural network, it is necessary to specify its architecture. Although the architecture may, to some extent, have been determined by the choice of a particular network, its details remain to be decided. For example, the Hopfield network has a single layer, fully connected architecture but the number of neural units must be specified. Having adopted a back-propagation network, it is necessary to determine the number of layers, the number of neural units in each layer and the number of feedforward connections.

The number of layers required depends on the complexity of the problem. For example, a single layer network can only distinguish between input classes that are linearly separable,

while three layer networks with sufficient neural units can separate classes of any arbitrary shape (Beale et al., 1990; Lippmann, 1987). The numbers of input and output elements depends on to the representation of the problem, while the number of hidden units tends be a somewhat arbitrary decision. Masters (1994) and Weigend (1994) indicate that choosing the appropriate number of hidden neurons is extremely important. Too few will starve the network of the resources it needs to solve the problem, Using too many will increase the training time and may cause *overfitting* (Weigend, 1994). The network will have so much information processing capability that it will learn insignificant aspects of the training set. Overfitting can also exacerbate noise in the *target* values (Moody, 1992; Reed et al., 1999). Masters suggests a rough guideline for choosing the number of hidden units following a geometric progression in each layer. A three-layer network with n input neurons and m output neurons would have $\sqrt{mn}$ hidden neurons (see *Figure 5.1*).

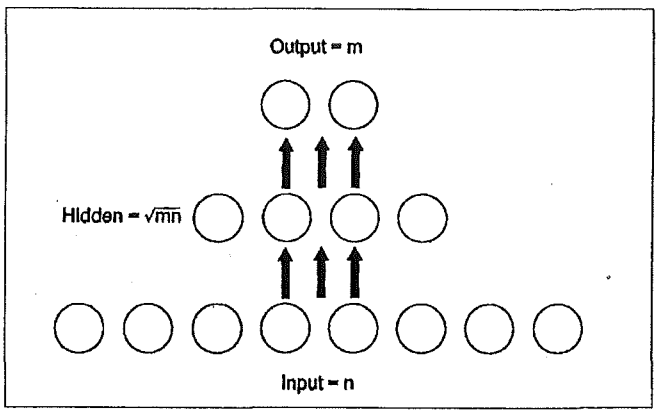


***Figure 5.1* – An approximate method for determining the number of hidden layer neurons.**

A similar rule can apply to four-layer networks, in which case the computation becomes:

$$r = \sqrt[3]{\frac{n}{m}}$$

$$\text{first hidden layer} = mr^2$$

$$\text{second hidden layer} = mr \qquad (5.1)$$

Sarle (1998) is very critical of these "rules of thumb" and states that there is no relevance to determining a good architecture based just on the number of inputs and outputs with a square root 'somewhere'. In his opinion it depends critically on the number of training cases, the amount of noise, and the complexity of the function or classification being learned.

Important is that the more hidden units in a network with one hidden layer - and hence the more dimensions in the weight space - the less the chance of being trapped in a bad local optimum when the network is being trained. The fact that increasing the number of hidden units reduces the risk of bad local optima is one of the most important properties of neural networks; without this property, it would indeed be impractical to use neural networks for complicated problems.

While this stresses the importance of the number of neurons, there are other researchers (Geman et al., 1992; Bartlett, 1997) that tout the size of the *weights* as being the most significant factor in determining good generalisation. Geman et al. found that excessively large weights leading to the hidden units could cause the output function to be rough, possibly with near discontinuities, or when leading to output units could cause wild output values far beyond the range of the data.

possibly with near discontinuities, or when leading to output units could cause wild output values far beyond the range of the data.

A lot of effort has been devoted to the design of *self-pruning* (destructive) or *incremental adding* (constructive) networks in order to arrive at either a smaller or a larger network, having initially started with the converse version (Saunders et al., 1996). A wide variety of methods have been proposed (Sietsma et al., 1988; Karnin, 1990; Fakhr et al., 1992; Chen et al., 1993, Omlin et al., 1993; Mozer et al., 1996), but most have in common the simultaneous minimisation of output error and minimisation of the number of hidden neurons. Many of these methods define an auxiliary criterion based on the number of hidden neurons, or on the size of the weights connecting hidden neurons to other layers (Weigend et al., 1991; Setiono, 1997), or on hidden-neuron activation distributions (Bishop, 1995). The function that is optimised is a composite of the output error with the hidden-neuron economisation criterion.

Although in a typical back-propagation network neural units are connected to all units in the following layer, this is not a mandatory requirement. Additional feedforward connections may be included between input and output layers, and connections between consecutive layers may be restricted.

In ANNs the choice of input representation as well as the structure of the network, e.g., the number of hidden units and the connectivity between layers, represents *a priori* knowledge in the network, because it puts constraints on the relations that the network can learn. In the last few years it has been shown that *sparsely* connected ANN architectures can be used to promote the training of networks with a large number of hidden units. The results of Ström (1997a, 1997b) for instance, indicate that increasing the number of hidden units is more important for the network's performance than to fully connect between the layers. With a sparse connection scheme between the input units and the hidden units, the generalisation of the network can be controlled by the connectivity rather than by smoothing the input representation.

McClelland (1989) trained a back-propagation neural network to predict which side of a balance beam would fall. The architecture of the network was such that the weight and distance information were analysed separately, before being combined (*Figure 5.2*). In fact, when neural units were fully connected to those of the following layer, the network did not perform the task as well. Thus, the architecture of the network imposes constraints on the training process, which may facilitate learning and generalisation if appropriate to the task (McClelland, 1989).

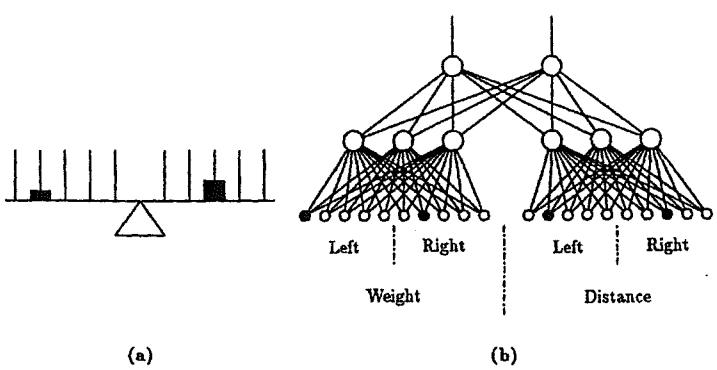


Figure 5.2 - **The balance beam task where the network is required to predict which side of the beam will fall. (a) Balance beam and (b) Network architecture.**

All practical applications of neural networks critically depend on the way the problem is represented (Anderson et al., 1989). Inputs to a neural network can take a variety of forms, such as raw data, features extracted from raw data, or some combination. For example, a network for time-series prediction may be presented with raw data consisting of sequential sections (windows) of a number of *different* historical time-series, including the series being modelled (Rattray et al., 1994).

Most neural networks perform better if they are provided with some ideas about the problem (Sarle, 1998). Although the way to solve a problem may not be known, some features that are likely to be helpful may be known, and this *a priori* information should be supplied to the network. Pre-processing can also be used to reduce the complexity of the task to be solved by the neural network. Consider the problem of recognising hand written characters drawn on a digitiser tablet

A high-resolution binary image (2048x2048 pixels, say) is obtained, where all pixels are zero except those through which the character passes (*Figure 5.3*). The vertical and horizontal extents of the character can easily be determined and the character placed in a box. This box can then be subdivided into an array of 15 x 10 smaller boxes.

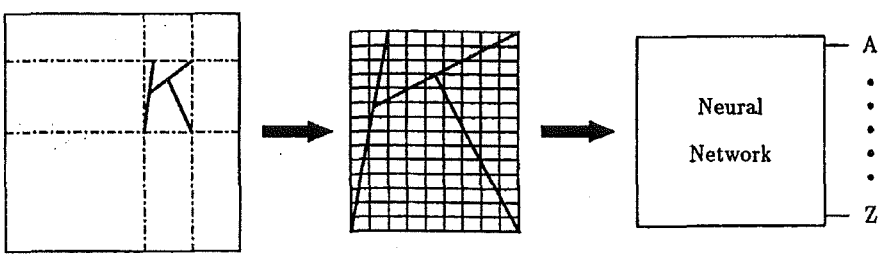


***Figure 5.3* - Pre-processing of characters so that the input vector is independent of scaling and translation.**

A binary vector **x** consisting of 150 elements can then be constructed, with $x_i = 1$ if the character passes through the *i*th box and $x_i = 0$ otherwise. This vector can be used as the input to a neural network. Thus, the neural network does not have to be insensitive to scaling or to translation.

Most neural networks require *normalised inputs*. This normalisation can be performed on individual input channels or across several input channels. Eberhart et al. (1990) have found that normalising related inputs (e.g., all amplitude measures) as a group is usually the most effective approach. Masters (1994) found through practical experience that some neural networks are limited in the range that their output can attain. For example, the feedforward network with logistics activation function is theoretically limited to an output range of 0.0 to 1.0. In practice, the range is more like 0.1 to 0.9 at most, with even these values being somewhat difficult to attain. A few networks, such as common versions of the Kohonen model, are limited in the values their inputs can take on. And virtually all networks train more efficiently if their inputs and outputs are restricted to a "reasonable" range.

This has important implications for *time-series prediction*. It means that a network's input must be scaled down to meet the neuron transfer function limitations, then conversely scaled up for the actual output, the predictions. There is another consideration; many common network models employ sigmoid activation functions. These squasher functions tend to emphasise the importance of intermediate output values, while obscuring fine differences when the outputs are near their extreme high and low values. This means that predictions that

approach the limits of the network's output will be less accurate than intermediate predictions. In particular, predicted values near the high end of the range would usually underestimate the best prediction, while those near the low end will overestimate the correct prediction. Thus the range of predicted values will be compressed relative to what it really should be.

A solution would be to scale the series to a narrow range near the *centre* of the network's limits, or employ *linear* output activation functions. But this can be counterproductive. In linearising the network, it can hamper its ability to recognise patterns. This is not an extremely serious problem, as hidden layers can still operate over their full non-linear range, which is all that is mathematically required. It is a consideration, though (Sarle, 1999).

Another problem with restricting the range or linearising the outputs is that immunity to *outliers* is compromised. The squashing activation functions are largely responsible for neural networks' robust behaviour in the presence of unusual noise. Reducing the range of data expands the range that can be covered before squashing takes place. Masters (1990) found that in some cases the results were excessively wild predictions.

Sarle (1998) adds his opinion to this important topic by stating that, rather than normalising the range of the input values to an obligatory [0, 1] interval, it is better to ensure that the values are *centred* around zero, allowing a greater range for the values if the interval is [-1, +1]. This type of approach is supported by Schraudolph (1998).

Neural networks learn from examples and, therefore, sufficient data must be available both to train the network and to evaluate its performance. The number of examples required to train a network successfully is very dependent on the network architecture, learning algorithm and specific application. The training set implicitly contains information on how to generalise (Hinton, 1989). Therefore, to ensure that the network does not learn any quirks of the training examples, large training sets should be used. In particular, there should be at *least* 4 times as many training examples as there are network weights to encourage generalisation rather than memorisation (Masters, 1994).

If the neural network is going to be effective at its ultimate task, the training set must be complete enough to satisfy two goals:

1. **Every class must be represented**. Usually, the training data will consist of several possible subgroups, each having its own central tendency towards a particular pattern. All of these patterns must be represented.
2. **Within each class, statistical variation must be adequately represented**. The presence of random noise imposed on pure patterns place a greater demand on achieving a good performance. A designer must ensure that an adequate variety of noise effects is included.

Eberhart et al. (1990) found that, for back-propagation networks, at least 10 examples of each output class are required to train the network adequately. The performance of the network must be evaluated on data not used for training. The amount of test data necessary depends on the user requirements. For example, in order to specify the percentage of correct classifications to a precision of 5%, 20 examples of each output class must be included in the test set.

5.3 Application examples

Neural network applications can be divided into a number of categories: memory, optimisation, categorisation, control, data processing, predictive models and pattern

recognition. Although the examples presented in this section illustrate typical problems from each of these categories, the distinction between classes is not clear cut, and some would argue that most applications are, in fact, pattern recognition tasks.

5.3.1 Memory

Neural networks can operate as *associative memories*, learning to associate input patterns with given output patterns. There are two types of associative memory, the first of which is known as *hetero-associative*. An m-dimensional input pattern ξ_p is associated with an n-dimensional output pattern ζ_p. Thus, when an input pattern **x** is presented to the network, the output pattern ζ_p, whose key ξ_p most closely resembles the input pattern **x**, is produced.

The second type of associative memory is termed *auto-associative*. In this type of network, a number of patterns ξ_p (p= 1, ... *P*) are stored and the pattern recalled is the one that most closely resembles the input pattern **x**. Such a network is, therefore, able to restore an image contaminated by noise to its original condition. Furthermore, an auto-associative network is able to retrieve an entire memory from any subpart of sufficient size. Thus, these networks perform pattern completion and are often termed content-addressable memories. For example, the Hopfield network operates as *a content-addressable memory* (Hopfield, 1982). However, the performance of the Hopfield network deteriorates once the number of patterns stored exceeds $0.15N$, where N is the number of neural units (McShane, 1992). Consequently, the Hopfield network is not suitable for large-scale storage, but attempts continue to increase its storage capacity.

5.3.2 Optimisation Problems

Neural networks can be used to solve complex optimisation tasks in which the aim is to minimise a cost function of many independent variables (Haykin, 1993; Tank et al., 1987; Potvin, 1997). The classic optimisation task is the travelling salesman problem, where it is necessary to find the shortest route to visit N cities, finally returning to the starting city.

In such problems, the data determine the network architecture required. For example, the travelling salesman problem can be represented by a NxN matrix of neural units (*Figure 5.4*), where the rows represent the N cities and the columns the order in which they are visited. Thus, *Figure 5.4a* shows a path C, A, E, B, D. Using this representation, only one neural unit in each row and column can be active at any one time and, therefore, inhibitory connections are placed between units of the same row and of the same column. The distances between cities (normalised to lie in the range 0.0-1.0) are used to form an additional set of inhibitory connections between neural units of adjacent columns (*Figure 5.4b*). Because the salesman must return home, the columns at each edge of the matrix are also considered to be adjacent. A solution to the problem is obtained by initialising the network to an arbitrary initial state and allowing it evolve to a stable state. Although the network may not find the best solution, it rapidly finds a near-optimal route. Furthermore, because no search is required. The time taken to find a solution does not increase exponentially with N, the number of cities to visit.

5.3.3 Classification

Neural networks, such as the *self-organising map* or *SOM and adaptive resonance theory* or *ART* (Grossberg, 1988), that employ *unsupervised* learning algorithms (*supervised* versions of ART also exist) can be used to classify or cluster input data. For example, Kohonen (1990) presented a self-organising map with short-time spectra of continuous Finnish speech. The spectra were calculated every 9.83 ms using a 256 point fast Fourier transform, from

which a 15 component spectral vector was formed. All such spectra were presented to the network in the natural order of utterance. During training the neural units became tuned to the acoustic units of speech, known as phonemes.

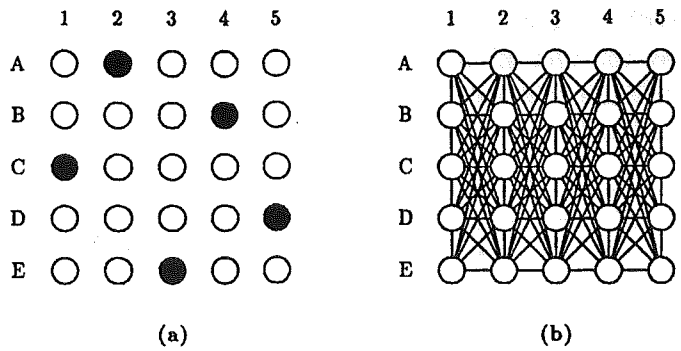


Figure 5.4 - **Travelling salesman problem: (a) network architecture with rows representing cities and columns the order in which they are visited and (b) inhibitory weights between neural units in the same row, in the same column, and in adjacent columns.**

Although the SOM is only a single layer network, it is able to discover features in the input data and display the relationships between them. Kohonen's original SOM does not optimise an 'energy' function (Erwin et al., 1995, Kohonen, 1995) and is not simply an information compression method like some of the other SOMs (Bishop, 1997).

The *probabilistic neural network* is an example of a *supervised* network that is also an intrinsic classifier (Specht, 1992, and Masters, 1994). It is suited to problems where the training set is large and training other models would be impracticably slow.

5.3.4 Control Problems

The control of systems with complex, unknown, and non-linear dynamics has become a topic of considerable research importance. In the design of *conventional* non-linear control systems, three main approaches have been used: i) adaptive control, ii) Lyapunov-based adaptive control, and iii) variable structure control (Jin et al., 1995). Indeed, the feedback linearisation technique of the non-linear systems is especially appealing from the point view of the non-linear control system design. To achieve the objective of either stabilisation or tracking, however, some strict assumptions were introduced regarding the structure of the uncertainties based on the completely known non-linear models.

The main potential of the neural networks for control applications can be summarised as, i) they can be used to approximate any continuous mapping to any desired degree of accuracy, ii) they perform this approximation through learning, and, iii) parallel processing and fault tolerance are easily accomplished.

One of the most popular neural network architectures used for control purpose is the feed-forward neural network with the error back-propagation algorithm. It is proved that a three-layered neural network using the back-propagation algorithm can approximate a wide range of non-linear functions to any desired degree of accuracy (Hecht-Nielsen, 1992; Simpson, 1990; Hornik, 1989). To avoid modelling difficulties, a number of multilayered neural networks based controllers have been proposed (Narendra et al., 1990; Hunt et al., 1992). Regarding a control system as a mapping of control inputs into observable outputs, an appropriate mapping is realised by a back-propagation network that is trained so that a

desired response is obtained. For such types of adaptive learning control systems, the neural networks are treated as subsystems of the whole control systems

For neural networks based non-linear control of a class of discrete-time non-linear systems, Narendra et al. (1991), and Hunt et al. (1991) first proposed an *identification*, or system modelling stage, and a non-linear *control* stage. The non-linear plant is identified off-line by the neural networks with the error back-propagation. The control is then initiated based on the model obtained by the identification. From this work, it has been concluded that for stable and efficient on-line control using the BP learning algorithm, the identification must be sufficiently accurate before any control action is initiated.

Solving such problems as balancing a broomstick on a cart (*Figure 5.5*) and reversing a trailer (*Figure 5.6*) provides examples of non-linear control. In a typical control problem, the state of the system $\mathbf{z}_{k+1}$ is a function $\mathbf{F}[\cdot]$ of the previous state $\mathbf{z}_k$ and the input $\mathbf{u}_k$:

$$\mathbf{z}_{k+1} = \mathbf{F}[\mathbf{z}_k , \mathbf{u}_k] \tag{5.2}$$

The aim is to provide the system with input vectors $\mathbf{u}_k$ that produce a desired state $\mathbf{z}^*$.

In many control problems, the input required to produce the desired state is not known. However, it is usually possible to provide the network with information regarding its performance. Consider the one-dimensional problem of balancing a broomstick that is free to pivot at its base connection on a moveable cart (*Figure 5.5*). Given information regarding the current state of the network (e.g., $\phi(t)$, $x(t)$), the force F to be applied to the cart in order to balance the broomstick must be determined. In general, the force required is not known but a measure of performance, such as the angle $\phi(t)$ that the broomstick makes with the vertical at time t, can be determined. Welstead (1994) used a feed-forward neural network as a controller system. Control is accomplished by having the network provide a look ahead to the next state of the system given the current state (Anderson, 1989). With this knowledge, appropriate control action is prescribed to assure that the new broom state will maintain a balanced system.

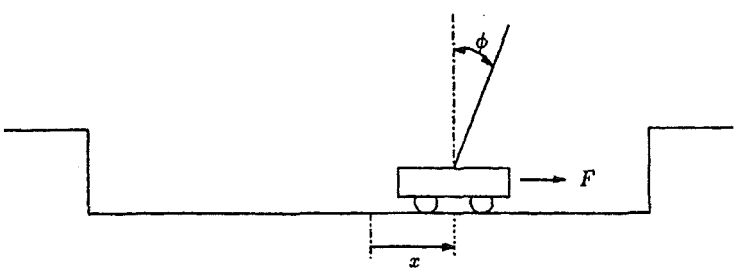


***Figure 5.5* - Balancing a broomstick on a movable cart.**

Tolat et al. (1988) trained a backpropagation network to balance a broomstick on a cart using training examples from human operators. However, because humans are unable to perform this task in real-time, a slowed down computer simulation of the system was used to obtain training examples. Given a sufficient period of training, the network learnt to emulate the human operator.

In other control applications the aim is to reach a final desired state $\mathbf{z}^*$ in some arbitrary time. For example, consider the problem of reversing a truck trailer up to a loading dock (*Figure 5.6*). The state of the system can be described by the position of the trailer (x_T, y_T), the angle ϕ_T of the trailer relative to the dock and the angle ϕ_C of the cab relative to the dock. From an

arbitrary initial state, it is necessary to reach the final state $\phi_T = 0^\circ$, $x_T = x_D$ where ($y_T = y_D$) describes the position of the loading dock. In order to train a back-propagation network an error signal is required at each time step, but in this situation only the error in the final state is known. Nguyen et al. (1990) overcame this problem by first training a two layer back-propagation network to emulate the system; given the current state and the input or steering signal the network learnt to predict the next state of the system. Having trained the emulator, another back-propagation neural network with 4 inputs (x_T, y_T, ϕ_T, ϕ_C), a single hidden layer of 25 units and one output unit was trained to produce an appropriate steering signal. The error in the final state was back-propagated through time using the emulator to determine the effect of a particular steering signal on the state of the system. Thus, an error signal and the appropriate weight changes at each time step were evaluated. Training took the form of 16 lessons, beginning with simple examples and progressing to arbitrary initial positions of the trailer. After training, the network was able to perform the task from an arbitrary initial condition (provided the trailer was a sufficient distance from the dock) with a root mean squared (r.m.s.) error in ϕ_T of 7° and in y_T of 3% of the truck length.

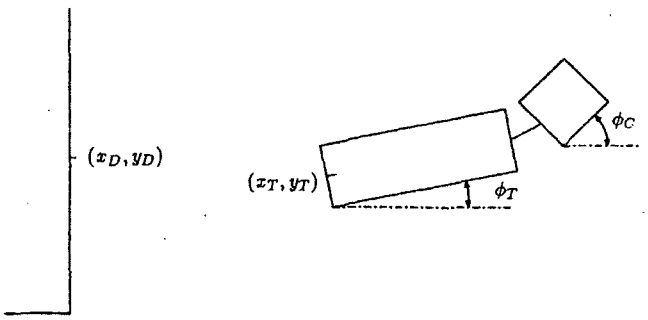


Figure 5.6 – **Reversing a truck trailer up a loading ramp.**

5.3.5 Data Processing

Traditional signal or data processing techniques tend to be linear (e.g., filtering and coding). Neural networks can be used to perform non-linear signal processing on structured signals, that is, signals that are always qualitatively similar, such as spectroscopy signals (Lee et al., 1992). However, because of the non-linearity of neural networks, the principle of superposition does not hold and, therefore, it is necessary to control the average power level of the input signal.

Noise removal techniques, for example, tend to be linear and rely on differences between the frequency content of the noise and that of the signal. However, when the frequency spectrum of the noise significantly overlaps that of the signal, the filtering process not only removes some of the noise but also modifies the signal itself. Neural networks can be used as non-linear filters and, if all signals to be considered are qualitatively similar, are able to remove noise that lies in the same frequency range as the signal. Hecht-Nielsen (1991) described a back-propagation neural network with a single hidden layer that was trained to filter ECG signals contaminated by white noise. Preliminary results produced signal-to-noise ratios which were 2-10 times better than those achieved using a traditional linear filter.

Neural networks can also be used to perform *non-linear data compression* on structured signals or images, as originally shown by Cottrell et al., (1987). Typically, a three-layer back-propagation network with m inputs, n hidden units and m outputs ($n < m$) is trained using the input pattern as the target output pattern. Once trained, the output of the hidden layer is a compressed form of the data. Dingle (1992) mentions the use of such a network to

compress ECG data before storage on tape. Prior to analysis the data were reconstructed using the weights between the hidden and output layers. Data compression rates of 1/15 - 1/100 were achieved with RMS errors in the reconstructed data of 0.1-0.5%. Masters (1994) comments that the principal problem with image compression (using block representation) is that faintly visible seam lines appear on the uncompressed image when the blocks are regularly spaced.

5.3.6 Predictive Models

Neural networks can be used as predictive models. For example, in the trailer reversing problem (5.3.4) a back-propagation network was used to model the behaviour of the system and predict its next state. Similarly, neural networks can be used to predict subsequent values of a time series. Elsner (1992) trained a back-propagation network to predict the next value of a chaotic time series generated from the Lorenz equation. The network was provided with the eight previous values of the time series. With three hidden units, the network predicted the next value of test data with an RMS error of 0.072, indicating that the network was capable of capturing the underlying chaotic dynamics.

In other applications, the network may be required to extract meaningful predictive information from a large amount of less valuable raw data. Consider the problem of scoring loan applications. Humans are particularly poor at this task because their decisions are too often influenced by factors such as appearance. Hecht-Nielsen (1991) described a back-propagation neural network to perform the task. The network was provided with relevant information from the loan application form, such as annual income, type of residence, length of residence, and produced a credit rating (e.g., dollar profit per dollar loaned per year). The training data were previous loans that had either been repaid or written off. Typically, 10,000 such examples are used to train the networks. Currently, many such networks are in regular use and perform significantly better than humans and statistical analysis techniques. It has been demonstrated that predictive neural networks are more advantageous than statistical techniques with low sample sizes and high levels of noise (Marquez et al, 1992). Arguments have also been put forward as to their advantage when building intelligent systems.

A more in-depth look at prediction type neural networks is taken in *Chapter* 6.

5.3.7 Pattern Recognition

Since neural networks are attempts to replicate some of the information processing capabilities of the brain, it is not surprising that many of their applications are in the field of pattern recognition (i.e., vision and speech processing). In these areas humans can easily outperform the fastest computers employing the most advanced algorithms. In fact, pattern recognition is one of the most promising fields for application of neural networks.

A recurring problem in hand-written digit recognition is the fact that the performance of the recognition system is highly test-set dependent. A system may successfully recognise 99% of the test data consisting of well-formed digits but score only 80% when confronted with the poorly formed digits that are both routinely produced and easily recognised by people.

Le Cun et al. (1992) used a 5-layer back-propagation network to recognise hand-written U.S. postal codes (*Figure 5.7*). It was recognised that classical work in visual pattern recognition had demonstrated the advantage of extracting local features and combining them to form higher-order features. This was accomplished in the network by forcing the early layers to perform two-dimensional convolutions over the 16x16-unit grey-scale input image. Layer

H1 accepted 12 individual 5x5-unit portions of the original image and allocated each portion to a 64-unit group, with a total of 768 hidden units. Layer H2 performed a similar process with different size portions. In total the network had a massive 1,256 units, 64,660 connections and 9,760 independent parameters. The MSE on the test set reached a minimum after 23 learning passes through the training set. The network was then saved and retrained for 5 passes using a data set that had undergone a slightly different pre-processing. The total number of training passes was therefore 28. Results indicated that 99.2% of the training patterns were correctly classified and 95.0% of the test patterns.

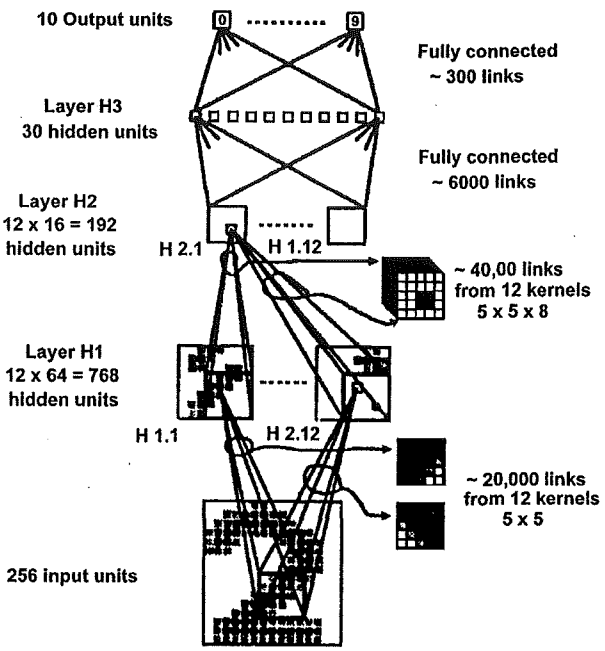


Figure 5.7 – **A five layer network architecture for recognising hand-written postal code digits.**

A simpler path was taken by Burr (1988) who trained back-propagation neural networks with a single hidden layer to recognise hand-written digits and letters. The digits were presented to the network in the form of a seven element shadow code. The network was trained on five examples of each digit and tested on a different five examples of each digit. Between 95-98% of the test set were recognised correctly depending on the number of hidden units. A 13 element shadow code was used to represent hand-written letters. The network was trained and tested on four examples of each letter and correctly classified up to 94% of the test examples.

Gorman et al.(1988) used a neural network to classify sonar returns from two undersea targets. A back-propagation network with one hidden layer was trained on sonar returns obtained from a variety of angles. The input to the network was a 60-element vector describing the power spectral density of the returns. The performance of the network tended to improve with increasing number of hidden units and 90% of test examples were classified correctly when 12 hidden units were used.

5.4 Practical issues in preparing input data

Different network models impose different constraints on their (training) data. It appears to be universally agreed that standardising/scaling/normalising input variables tends to make the training process better behaved.

Standardising input variables also has different effects on different training algorithms for multi-layered feed-forward networks. Sarle (1998) states that *steepest descent* is very sensitive to scaling. The more ill-conditioned the Hessian is, the slower the convergence. Hence, scaling is an important consideration for gradient descent methods such as *standard backpropagation*. *Quasi-Newton* and *conjugate gradient* methods begin with a steepest descent step and therefore are scale sensitive. However, they accumulate second-order information as training proceeds and hence are less scale sensitive than pure gradient descent. *Newton-Raphson* and *Gauss-Newton*, if implemented correctly, are theoretically invariant under scale changes as long as none of the scaling is so extreme as to produce underflow or overflow. *Levenberg-Marquardt* is scale invariant as long as no ridging is required. There are several different ways to implement ridging; some are scale invariant and some are not. Performance under bad scaling will depend on details of the implementation.

Two of the most useful ways to standardise inputs are:

- Mean 0 and standard deviation 1
- Midrange 0 and range 2 (i.e., minimum -1 and maximum 1)

Time-series data is typical of input data that are measured on an *interval scale*. Variables measured on an interval scale are almost always presented to a neural network using exactly one neuron. The variables must be scaled in such a way as to be commensurate with the model's neuron-activation limits. Care must be taken that the scaling is done so that the data used in training will be commensurate with that used in testing.

Probably the most common scaling method employed is simple linear mapping of the variable's practical extremes to the network's practical extremes. In the unusual case that a measured value goes beyond the limit, the value would be truncated to that limit.

Let the variable's maximum and minimum values expected in normal use be designated V_{max} and V_{min} respectively. Let the network's practical limits be designated A_{max} and A_{min}. For a feedforward network with logistic activation functions, output activation limits would typically be 0.9 and 0.1, respectively. Inputs have no theoretical limits, but stability is usually improved by using comparable limits. An observed value V is scaled to a presentable value A with:

$$A = r\,(V - V_{min}) + A_{min} \tag{5.3}$$

$$r = \frac{A_{max} - A_{min}}{V_{max} - V_{min}} \tag{5.4}$$

If the variable is used to train an output neuron, the activation levels need to be unscaled to obtain meaningful values for the variable. Inverting (5.3) gives:

$$V = \frac{(A - A_{min})}{r} + V_{min} \tag{5.5}$$

Measured variables often have an approximately *normal distribution*. If a variable is unimodal, nearly symmetrically distributed about its mean, and virtually never has significant values extremely far from its mean, a more sophisticated normalisation based on its mean and standard deviation can be employed. A random variable can be standardised to a *Z-score* by subtracting its mean and dividing by its standard deviation:

$$Z = \frac{x - \mu}{\sigma} \tag{5.6}$$

This removes all effects of offset and measurement scale. If for instance a set of objects were weighed twice, first in kilograms and then in pounds, then scaling to a Z-score would reduce both sets of measurements to comparable units. Masters (1994) notes that in practice, this scaling will often turn out to be almost identical to that provided by equation 5.3.

If a theoretical basis for knowing the mean and standard deviation of a variable is not available then the parameters can be estimated from a sample:

$$\mu = \frac{1}{n}\sum_{i=0}^{n-1} x_i \tag{5.7}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n-1}(x_i - \mu)^2} \tag{5.8}$$

Division is by n - 1 rather than n in the above equation as the standard deviation of a population *sample* is being estimated. The exact sample standard deviation, obtained by dividing by n, would, on the average, slightly underestimate the true population standard deviation.

Scaling to a Z-score is generally not sufficient, as the scaled values would still exceed the activation limits implicit in many neural network models. After Z scaling, equation (5.3) is applied to the practical limits of the Z-scores to bring values in line with what the network demands. It should be noted that Z-scores have no theoretical bounds and can approach positive and negative infinity. However, if the variable is approximately normally distributed, the Z-scores definitely have practical limits. *Table 5.1* shows the probability of the absolute value of a Z-score exceeding some limits.

Z limit	P(exceeding)
1.28	0.20
1.64	0.10
1.96	0.05
2.58	0.01

***Table 5.1* – Probabilities of the Z-score limits being exceeded.**

These "two-tailed" probability figures are the probability that a Z-score will either be greater than the limit or be less than the negative of the limit. When choosing an acceptable probability the table will indicate the corresponding practical limits. V_{max} would be set equal to the value shown, and V_{min} equal to the negative of it, using equation (5.3) to do the final scaling from Z-score to neuron activation.

The choice of a comfortable limit from the above table involves a trade-off between the usable range and the probability of having to truncate values. A large limit equates to a small probability that an observed value will exceed this limit and have to be truncated at the neuron's activation limit. On the other hand, the majority of observed values would be scaled into a relatively narrow range, limiting the diversity of values presented to the network. This can impede learning. Conversely, picking a small limit will allow the full range of activations to be attained for most observations. The latter case is thought to be preferable when the variable is presented to neurons having *firm* activation limits.

The converse is true when applying data to the *input layer* of a feed-forward network, as this usually has no firm activation limits (i.e. when using the identity function). In order to avoid information loss a large limit is preferred. The network will make up for the compressed range when its input weights are learned. If the network does not impose firm limits, the second scaling can be avoided altogether, thus using Z-scores directly as inputs

Z-score scaling and neuron-limit scaling can be combined into one operation:

$$A = r\left(\frac{x-\mu}{\sigma} - V_{min}\right) + A_{min} \tag{5.9}$$

where r in the above equation is as defined in equation (5.4), noting that V_{min} and V_{max} in that equation and the above equation are from the previous table of Z-score limits, not referring to the raw data itself. The above formula is inverted to convert output-neuron activations to original variable scales:

$$x = \frac{\sigma}{r}A + \left(\mu + \sigma\left(V_{min} - \frac{A_{min}}{r}\right)\right) \tag{5.10}$$

The procedure of scaling by both standard deviation and neuron limits will in practice usually give results that are nearly identical to those obtained with the much more straightforward procedure of scaling based on practical data limits. This is because the pragmatic limits will usually be close to the limits implied by the variable's mean and standard deviation, resulting in the same ultimate transformation constants.

There are, however, two reasons for utilising the more complex method (if the variable's distribution allows it):

1. It removes the arbitrary nature of choosing limits for the variable. The limits are selected by a more objective criterion, although the choice of the probability limit is somewhat arbitrary.
2. Scaling of variables can be easily *automated.* Rather than requiring manual input in the choice of limits, automatic scaling based on mean and standard deviation can be incorporated into the computer program.

Especially point 2. complements the automated data gathering/processing approach of the desired fluid energy management system (FEMS).

When a set of input variables is suspected of having an *uneven* distribution, it may be more difficult for a neural network to learn to use them, even if the values are linearly scaled to a reasonable range. This is because the information content in the variable is too distorted. Small but important variation may be compressed into a relatively narrow area, while other variation is spread out in a wider range than its importance justifies. In such a case a *non-linear transformation* might prove necessary.

Iglewicz (1983) alludes that there are three properties that the input data should possess. It is the goal of non-linear transformation to try and endow the data with these properties if they are found lacking, although they cannot always be all satisfied. The desirable properties are as follows:

Homoscedasticity - The variance of the data should be approximately the same for all values it takes on. A variable with a large value and with a high variance is detrimental.

Normality - A normal distribution is not particularly important to a neural network. In fact, there is some evidence that flat distributions are learned most easily (Masters, 1994). What is

important is that the distribution be approximately symmetrical and not have a heavy tail (i.e. frequent wild values).

Additivity - Most practical neural networks have more than one input variable. It helps the network to learn if the contributions of these variables are as additive as possible. When, for example, it is the product or quotient of two variables that is important to the decision, than the network is being burdened more than needed. Multiplicative relationships can be changed to additive by taking logs.

Neural networks are not nearly as sensitive to non-standard distributions as most traditional techniques are. They can work with variables having distributions that standard linear models, i.e. multiple regression, could not possibly handle

5.5 Limitations of neural networks

In principle, neural networks can compute any computable function, i.e. they can do everything a normal digital computer can do. However, neural networks are particularly poor at arithmetic and formal logic, and they are not very good at storing and retrieving very large amounts of data with high accuracy. It is also equivocal what can and cannot be learnt by a given network, and what architectures are required for the acquisition of given sorts of skills.

In practice, neural networks are especially useful for classification and function approximation/mapping problems which are tolerant of some imprecision, which have lots of training data available, but to which hard and fast rules (such as those that might be used in an expert system) cannot easily be applied. Almost any mapping between vector spaces can be approximated to arbitrary precision by feedforward neural networks, which are the type most often used in practical applications.

Feedforward networks with a single hidden layer are statistically consistent estimators of regression functions, under certain practical assumptions regarding sampling, target noise, number of hidden units, size of weights, and form of hidden-unit activation function. Such networks can also be trained as statistically consistent estimators of derivatives of regression functions (White et al., 1992). Feedforward networks with a single hidden layer using threshold or sigmoid activation functions are universally consistent estimators of binary classifications (Lugosi et al., 1995; Devroye et al., 1996) under similar assumptions.

Unfortunately, the above consistency results depend on one critical assumption: that the networks are trained by an error minimisation technique that comes arbitrarily close to the global minimum. A continuing topic of research is to find the learning algorithm that will ensure this with minimal computation.

Neural networks are, at least today, difficult to apply successfully to problems that concern manipulation of symbols and memory. And there are no methods for training neural networks that can miraculously create information that is not contained in the training data. It is also unable to simulate human consciousness and emotion. Artificial neural networks may be useful for modelling some aspects of or prerequisites for consciousness, such as perception and cognition, but on the whole consciousness is still one of the world's great mysteries.

Chapter 6. Discrete Time Series Prediction using Neural networks

6.1 Introduction

Time series are sequences, either discreet or continuous, of quantitative data. Simple series consist of a single value or observation at each instant of time, whereas multiple series comprise a set, or vector, of different observations at each instant.

Time series prediction has long been the domain of statisticians. The importance of this topic is reflected by the diversity of its applications in different business sectors, including industrial process control, commodity demand prediction and financial market forecasting. The prediction involving real processes must adapt to ageing and other changes occurring within the system, in addition to a myriad of other external influences, many of which are unforeseeable. Hence, in general, prediction cannot give an accurate estimate of the future. However, a system that is able to learn and adapt with time is better able to predict the most likely future behaviour.

Prediction techniques can also be useful in determining the underlying behaviour of a system from a time series of measurements. More specifically, they can be used to distinguish between noise and apparently random data which are, in fact, chaotic activity generated by a deterministic system.

The classic approach to time series predictions is to carry out a manual analysis of the time series data, build a model from first principles and then iterate that design by measuring the closeness of the model to the real data. This can be a long process, often involving the derivation, implementation and refinement of a number of models before one with appropriate characteristics is found. In addition, problems arise when the underlying dynamics of the system are poorly understood. In particular, the most difficult systems to predict are those with non-stationary dynamics, where the underlying behaviour varies with time. Problems also arise because physical data are subject to noise and experimental error. Some series are also relatively short, providing few data points on which to conduct the analysis. In all these situations (where the data set is small and/or noisy, a good model is lacking, the system behaviour is non-stationary), traditional techniques are severely limited and alternative techniques must be found.

Artificial neural networks offer such an alternative approach. They are at their most powerful when applied to problems whose solutions require knowledge that is difficult to specify but for which there is an abundance of examples. As time series prediction is performed entirely by inference of future behaviour on examples of past behaviour, it is an ideal application for neural network technology.

The most popular method for the teaching of neural networks is the back-propagation algorithm as described in *Chapter* 4. However, a major limitation of the standard back-propagation algorithm described there is that it can only learn an input-output mapping that is *static*. This form of mapping is well suited for pattern-recognition applications, where both the input vector **x** and the output vector **y** represent *spatial* patterns that are independent of time. The back-propagation network is, however, able to perform non-linear prediction on a *stationary* time-series, i.e. one whose statistics to not change with time (Haykin, 1994). In such a case the standard feed-forward network would have inputs consisting of past samples of the time-series. Typically the output vector **y** would be a single prediction.

In order to deal with time-varying forms of data the static model of the neural network needs to the altered to allow time to be represented by the effect it has on data processing. This means providing the mapping network dynamic properties that make it responsive to time-varying input. Elman (1990) suggested that for a neural network to be dynamic, it must be given memory. Typically this is achieved by the introduction of time delays in the structure of the network, which are subsequently altered during the training phase. Adaptable delays are a mechanism through which neural systems can tailor their own dynamics. Evidence from neurobiology indicates that delays, especially if adaptive, are useful in information processing (Baldi et al., 1994). Baldi showed that when delays are introduced in networks it influences their stability and the output from each neuron layer can become oscillatory, the exceptions being feed-forward networks or networks with very small gains and/or synaptic weights.

Since the late '80's a number of networks and learning mechanisms have been introduced that use time-delay in one form or another. Two of the most popular structures are the *time-delay neural network* (TDNN), as first described by Lang et al. (1988), and the *recurrent neural network* (RNN). The latter has a number of learning approaches; two of the better known algorithms are: BPTT - *Back-Propagation Through Time* (Werbos, 1990) - introduced by Paul Werbos in a 1974 PhD thesis, RTRL - *Real-Time Recurrent Learning* (Williams et al., 1989) - published by messrs.Williams and Zipser. Haykin (1994) eludes that the origin of the latter approach can be traced to an earlier paper by McBride et al. (1965) on system identification for tuning the parameters of an arbitrary dynamic system. Other learning methods are *Recurrent Back-Propagation* (Pinenda, 1988) and *Dynamic Back-Propagation* (Narendra, 1991). BPTT and RTRL especially, have a significant number of variants. This chapter focuses on these methods with a view to determining the most suitable structure for FEMS.

6.2 The Time-Delay Neural Network

Time-Delay Neural Networks (TDNN) were originally developed in the context of speech recognition to combine pattern learning and alignment. It is a multi-layer feed-forward network whose non-recurrent architecture (*Figure 6.1*) utilises past as well as present inputs. During training, the alignment procedure automatically defines significant features in the sequences in such a way as to allow recognition on the basis of multiple features and their relative positions. With this type of architecture the various time delays can be introduced between the input and hidden layer(s) and, if desired, the hidden and output layers.

Lang et al (1989) devised a network to recognise the four single letters 'V', 'E', 'D', and 'B', pronounced as words, each represented by an equivalent two-dimensional 12 x 16 unit spectrogram which was obtained after special processing. The network architecture is based on providing multiple copies of each output unit. The copies of an output unit apply the same weight pattern to successive narrow slices of the input pattern, attempting to locate a sub-pattern which is characteristic of the word denoted by that unit.

A two-layer version of this network is shown in *Figure 6.3*, where the output units have been connected to narrow receptive fields that only cover 5 time steps.

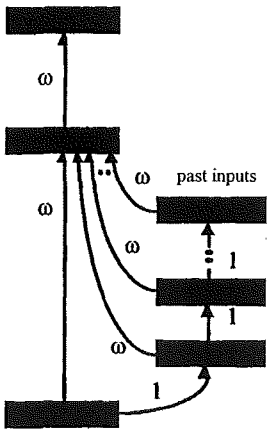


Figure 6.1 – **Architecture of a three-layer Time-Delay Neural Network (TDNN): ω signifies trainable weights, 1 signifies that the activations at the destination are a copy of the activations at the source in the previous processing cycles.**

Since there are eight ways to position a 5-step window on a 12-start pattern, the network contains 8 copies of each output unit. When an input is presented to the network, each of the 4 x 8 = 32 output unit copies is activated by an amount that indicates the copy's confidence that its word is present, based on the evidence that is visible in its receptive field.

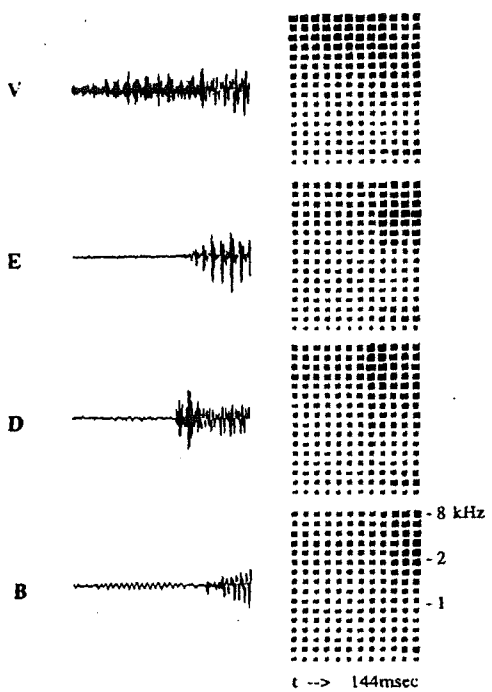


Figure 6.2 - **The waveforms shown are 144msec slices extracted from the 4 words 'V', 'E', 'D' and 'B', together with their equivalent 12x16 input layer spectrograms.**

The same concept was applied to a more powerful three-layer network. This had 192 input units encoding the word spectrogram. The hidden layer contained 10 copies of 8 hidden units that were each connected to 3 frames of the input, depicting time-delays of 0, 1 and 2. The third layer had 6 copies of the 4 output units, each looking at 5 frames of the pseudo-spectrogram generated by the hidden layer (*Figure 6.4*), thus depicting time-delays of 0, 1, 2, 3 and 4. A recognition score of 93% was obtained on test data different from the training

data. In a more elaborate study reported by Waibel et al. (1989), a TDNN with two hidden layers was used for the recognition of three isolated words: 'B', 'D' and 'G'.

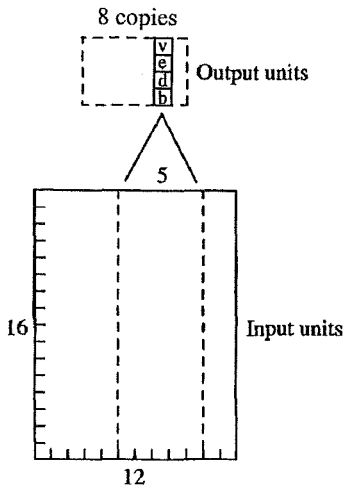


Figure 6.3 – **A two-layer network whose output units are replicated across time.**

In performance evaluation involving the use of test data from three speakers, the TDNN achieved an average recognition score of 98.5%. Waibel et al. reports that the power of the TDNN lies in its ability to develop shift-invariant internal representations of speech and to use them for making optimal classifications.

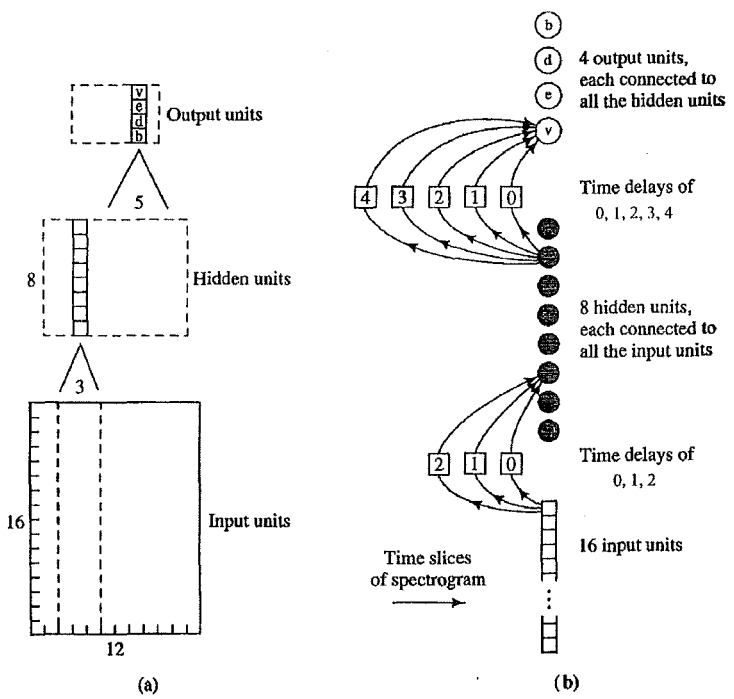


Figure 6.4 – **(a) A three layer network whose hidden units and output units are replicated across time. (b) Time-delay neural network (TDNN) representation.**

Training the network was achieved by the standard back-propagation algorithm. Wan (1994) represented the TDNN topology with a multi-layer perceptron (MLP) in which each synapse takes the form of a *finite-duration impulse response filter* (FIR); as such the network is referred to as an FIR-MLP. It is a feed-forward network and attains its dynamic behaviour by virtue of the fact that each synapse of the network is designed as an FIR filter (Haykin,

1994). The back-propagation algorithm can also train this network, but a computationally simpler *temporal back-propagation algorithm* exists and was first described by Wan (1990).

The idea of replicating network hardware to achieve position independence is not new (Fukushima, 1980). Replication is especially common in connectionist (i.e. neural network) vision algorithms where local operators are simultaneously applied to all parts of an image (Marr and Poggio, 1976). Lang et al. write that the inspiration for the external time integration step of their time-delay neural network (TDNN) was Michael Jordan's work on back-propagating errors through other post-processing functions (Jordan, 1986).

Waibel (1989) describes a modular training technique that made it possible to scale the TDNN technology up to a network which performs speaker dependent recognition of all Japanese consonants with an accuracy of 96.7%. The technique consists of training smaller networks to discriminate between subsets of the consonants, such as *bdg* and *ptk*, and then freezing and combining these networks along with "glue" connections that are further trained to provide interclass discrimination.

Other researchers have independently designed networks similar to the TDNN. The time-concentration network of Tank et al. (1987) was motivated by properties of the auditory system of bats, and was conceived in terms of signal processing components such as delay lines and tuned filters. This network is interesting because variable-length time delays are learned to model words with different temporal properties, and because it is a neural network speech recognition system implemented with parallel hardware instead of being simulated by a serial computer.

In the TDNN the time delays are fixed throughout training and strong weights evolve for interconnections whose delay values are important to the pattern classification task. Lin et al. (1993) presented an *adaptive* time delay neural network (ATNN) that adapted not only the connection strengths but also the time delay values during training, to better accommodate to the spatio-temporal patterns. They generalised the *temporal back-propagation* algorithm described in *Section* 6.4 to permit the time delays to be adapted in a continuous fashion. The effectiveness of the TDNN was demonstrated on a chaotic time-series prediction.

6.3 Further predictive applications of Time-delay neural networks

There are a large number of connectionist systems that utilise the TDNN as a prediction mechanism. The greater portion (published by the better known authors) appear, perhaps for historical reasons, to operate in speech recognition processes. But other successful applications have also been found in the financial and the computer industries. A more recent example is in gene-modification research where a TDNN is used for gene location prediction, as explained in *Section* 6.3.2.

6.3.1 Stock Market Prediction

Tan et al. (1995) describes using a *Time Delay Neural Network* (TDNN) techniques to predict significant short-term price movement in a single stock, Apple Computers, on the Stock Market. Having pre-processed the data, a *Probabilistic Neural Network* (PNN) was trained to predict *trends*. The primary objective was to limit false predictions (known in pattern recognition literature as *false alarms*). False alarms are worse than missed opportunities, because they lead to losses if acted upon. A false alarm rate of 5% was achievable with the correct system design and parameterisation. A TDNN was utilised in predicting the *actual price* increase over the period of the next month (only price gains large enough to create a reasonable profit opportunity were registered as being significant). The

network was trained to minimise an error measure for multiple-step-ahead predictions of the price itself. The best results were obtained by applying an *expanding window* strategy. First, the weights of the TDNN were adapted to minimise one-step-ahead prediction errors. Then the total error for two-step-ahead predictions was minimised using the previous final weight values. This process was iterated until the error for the final 22-step-ahead predictions had been minimised.

From the TDNN results the 2% upward trend could also be derived. Direct prediction of the trend by PNN, instead of iterative price forecasts, appeared to be easier, but the best results were achieved by the TDNN. Tan et al. speculated that this was because the trend prediction involved architecture modifications that were unnecessary in price prediction; presumable this is detrimental to the forecasting accuracy.

6.3.2 Gene end-points prediction

Mache et al. (1998) used a three layer Multi-State Time-Delay Neural Network as a reliable algorithmic means for the detection of genes in unknown human genomic sequences. The biggest weakness of currently available alternative algorithms is the detection of the *gene endpoints*. If gene endpoints are not correctly found the gene sequences often have too many or not enough exons, or accidentally combine two genes. The network was trained specifically to detect the so-called POL II promoter regions in DNA sequences

Standard Time-Delay networks can be extended by *additional* state layers to Multi-State Time-Delay neural networks (MS-TDNN) (Bodenhausen et al., 1991). Multi-State TDNN's allow the recognition of ordered sequences of features independent of variations in their relative positions.

The ability of MS-TDNNs to combine pattern learning and alignment means that during training, the alignment procedure automatically redefines significant features in the sequences in such a way as to allow recognition on the basis of multiple features and their relative positions. The network was able to detect the elements despite major variations in their relative positions on the DNA sequence. It had the ability to detect combinations of sequence elements in a set of larger sequence fragments without initial knowledge of the location and sequence characteristics of the elements.

A Dynamic Time Warping algorithm (Sakoe, 1987) was used to find the optimal alignment path that maximised the (sum of) scores of the sequence elements. The network was trained with 640 randomly chosen vertebrate. For the validation of the network, Mache et al. used 35 different promoter sequences that were not included in the training set. A threshold value was optimised by scanning four human genes from GENBANK database. For the given results they tested a set of 74 human genes with this independent chosen threshold value. Unlike the alternatively utilised algorithms in this area of research, TDNN was able discover some biologically significant features ab initio, a feature that is expected to be valuable in, for example, learning to recognise particular subclasses (often poorly characterised experimentally) of POL II promoters. In its reported implementation, accuracy was of the same or better order as that of other algorithms based on much more prior knowledge. As an indication of just how challenging this area of application is for neural networks, Mache et al. report that in a test set of 74 genes, 58% of true promoters were recognised. This was considered a good result.

6.3.3 Predicting multiprocessor memory access patterns

That neural network techniques are also applicable to computer system optimisation is shown by Sakr et al. (1997) whom use three different *on-line* prediction methods to forecast the *memory access patterns* for multiprocessors that are forced to share available memory.

Large-scale multiprocessor systems require low-cost, highly scalable, and dynamically reconfigurable *interconnection networks* (INs) because completely connected networks are highly complex and suffer from soaring costs. Accordingly the standard INs offer a limited number of communication channels that are configured on demand to satisfy required processor-memory accesses. In this demand driven environment, a processor accessing a memory module makes a request to an IN controller to establish a path (reconfigure the IN) that satisfies the processor's request. The controller is used to optimise the required IN configuration based on the set of current processor requests. Hence, the end-to-end latency incurred by such INs can be characterised by three components (*Figure 6.5*): *control time*, which is the time needed to determine the new IN configuration and to physically establish the paths in the IN; *launch time*, the time to transmit the data into the IN; and *fly time*, the time needed for the message to travel through the IN to its final destination. Launch time can be reduced by using high bandwidth opto-electronic INs, and fly time is relatively insignificant in such an environment since the end-to-end distances are relatively short. Therefore, control time dominates the communication latency.

However, in a multiprocessor system executing a parallel scientific application, the memory-access requests made by the processors follow a *repetitive pattern* based on the application. Compilers can analyse an application and attempt to predict its access patterns (Gornish, 90), but often the pattern is dynamic and thus hard to predict. The goal of Sakr et al.'s work was to employ a technique that learned these patterns on-line, predicted the processor requests, and performed the IN configuration prior to the requests being issued, thus hiding the control latency. The effect is a significant reduction in the communications latency for multiprocessor systems.

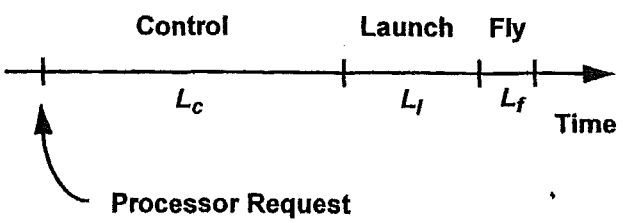


***Figure 6.5* - The three components of the end-to-end communication latency; control time, launch time and fly time. Control time dominates overall communication latency**

Three different on-line prediction techniques were tested to learn and predict repetitive memory access patterns. These were i) a Markov predictor, ii) a linear predictor and iii) a time delay neural network (TDNN) predictor. Each prediction method was trained using the access patterns for three typical parallel processing applications, the 2-D relaxation algorithm, matrix multiply and Fast Fourier Transform. Much as expected, different predictors performed best on different applications; however, Sakr et al. concluded that the TDNN produced the best overall results and hypothesised that the TDNN had the best chance of adapting to different memory access patterns from the variety of real applications. The question left open was how these prediction methods can be efficiently implemented in hardware.

6.3.4 Phoneme probability estimation

Nikko Ström (1997) presents a method for training large neural networks for phoneme probability estimation. An architecture combining time-delay windows and recurrent connections (much as implemented and discussed in *Chapter* 8) is used to capture the important dynamic information of the speech signal. Because the number of connections in a fully connected recurrent network grows super-linear with the number of hidden units, schemes for sparse connection and connection pruning are explored (Ström, 1997a, 1997b). It was found that sparsely connected networks outperform their fully connected counterparts with an equal number of connections. The implementation of the combined architecture and training scheme is described in detail. The achieved phoneme error-rate, 27.8%, for the standard 39 phoneme set on the core test-set of a standardised database, is in the range of the lowest reported (all training, results and the simulation software used is made available by the author on the internet).

6.4 Temporal back-propagation learning

The standard back-propagation algorithm can only be used with the *static* model of the feed-forward network. To be able to deal adequately with temporal input and the delays introduced in the time-delay neural network, it is necessary to unfold the network *in time*, thereby removing the delays and creating an equivalent but larger 'static' version of the network. This can then be trained with the standard back-propagation algorithm.

To unfold the network in time, it is possible to proceed in two ways (Haykin, 1994), *Forward unfolding in time* - starting at the input layer and move forward through the network, layer by layer, and *Backward unfolding in time* - starting at the output layer and move backward through the network, layer by layer. In both cases, there is growth in the size of the network that results from unfolding it in time. In the forward case, the growth is of order $o(dn)$, where d is the total number of time delays and n is the total number of free parameters in the network. In the backward case, the size of the equivalent static network grows *geometrically* with the number of time delays and layers. Forward folding in time is therefore preferred over backward folding in time.

The major disadvantages of this method (Wan, 1990) are that it lacks a recursive formula for propagating the error terms and that there is a need to keep track of which static weights are actually the same in the equivalent network.

Wan overcame these problems by considering an alternative way of expressing the partial derivative of the error function E_{total} with respect to the weight vector as seen previously in equations (4.23) and (4.24);

$$\frac{\partial E_{total}}{\partial \mathbf{w}_{ji}} = \sum_{n} \frac{\partial E_{total}}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial \mathbf{w}_{ji}(n)} \tag{6.1}$$

where E_{total} is defined as being the value of $E(n)$ computed over all time and n denotes discrete time and is known as the time index. The weight vector $\mathbf{w}_{ji}$ is made up of the values of $w_{ji}(l)$, $l = 0, 1, 2, \ldots, m$, where m is the total number of delays in each layer.

For equation (6.1) the time index runs over $v_j(n)$ and not $E(n)$. The partial derivative $\partial E_{total} / \partial v_j(n)$ can be interpreted as the change in error function E_{total} produced by a change in the net activation potential v_j of neural unit j at time n.

The weight-update equation for temporal back-propagation can be summarised as (Wan, 1990):

$$\mathbf{w}_{ji}(n+1) = \mathbf{w}_j(n) + \eta\, \delta_j(n)\, \mathbf{x}_i(n) \tag{6.2}$$

where η is the learning rate, and the explicit form of local gradient $\delta_j(n)$ depends on whether neuron j lies in the output layer or the hidden layer. $\mathbf{x}_i(n)$ is the input vector applied to neuron j at time n. Details of the derivation of the gradient $\delta_j(n)$ can be found in Wan (1990) and Haykin (1994).

6.5 The Recurrent Neural Network

The recurrent network to be discussed in this section and the time delay neural network (TDNN) presented in the previous sections integrate both spectral and temporal representations into a single ANN structure.

Recurrent neural networks are characterised by both feedforward and feedback paths. The feedback paths enable the activation at any layer to either be used as an input to a previous layer, or be returned to that layer after one or more time steps. One of the motivations for using recurrent networks for temporal modelling is that they allow time to be represented by the effect it has on processing a sequence (Elman, 1990). It differs from a Hopfield network (*Chapter* 4), which is also a recurrent network, in two important respects: i) the network has hidden neurons, ii) it has arbitrary dynamics. In fact, RNNs can be viewed as discrete-time dynamical systems (Tino et al., 1995) and the literature dealing with this relationship is quite extensive, (Casey, 1995a, 1995b; Beer, 1994; Blum et al., 1992; Hui et al., 1992) are but a few examples. The diversity of dynamic behaviours suggests that recurrent networks may well be suited to the problem of time-series prediction.

In most recurrent networks, the activation of the node in layer q at time t is stored in a *context node* so that it may be used as an input to some set of nodes (generally within the same layer) at time $t + 1$. The weights between the context nodes and the nodes they feed into are called *context weights.* In the simplest example, a node has a "self-transition" weight when the activation at time step t is delayed and then used as an input to determine the activation at time step $t + 1$. *Figure 6.6* illustrates that self-transition weights are a special case of context weights. That is, the self-transition weight connects to the node from which the activation originated, while the context weights are fully connected to the nodes in layer q.

Recurrent networks are appealing because they provide networks with *internal states* and a form of *memory.* As a result, recurrent networks provide more than the simple 1-to-1 mapping of feed-forward networks. The output of such networks depends not only on the current inputs but also on previous inputs i.e. they can integrate activity levels over time, and thus "remember" activations from previous inputs.

Elman (1990) introduced a particular class of recurrent network in which the feedback connections are from the state vector to the hidden layer, as illustrated in *Figure 6.7.* Elman used this neural network architecture, along with the backpropagation learning algorithm, to learn the grammatical structure of a set of sentences randomly generated from a limited vocabulary and grammar.

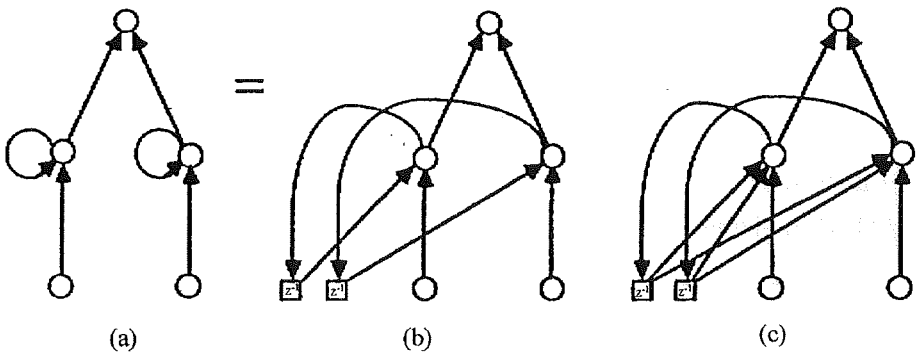


Figure 6.6 - **(a) A recurrent network with self transitions. (b) The temporal flow network redrawn with time-delayed context nodes (shown as small boxes with z^{-1}). (c) A network with a fully-connected recurrent hidden layer is represented using context nodes.**

A major point of Elman's work was to study the hidden unit activation patterns in a trained network, produced in response to a sequence of inputs, and to use techniques such as cluster analysis to infer a structure for the data as represented in the hidden unit activation patterns. He was able to extract a cluster hierarchy corresponding to the syntactic rules from which the data had been constructed: nouns, verbs, animate and inanimate nouns, transitive and intransitive verbs, etc. This information was only implicitly present in the data presented to the network: in other words, the network had learned the structure of the linguistic data from the examples presented to it.

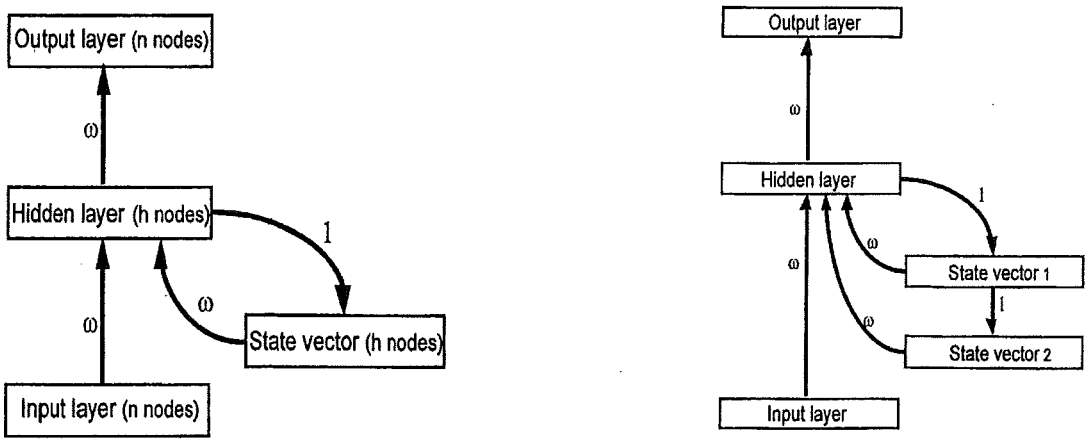


Figure 6.7 – **Architectures of an Elman recurrent network with one and two state vectors: ω signifies trainable weights, 1 signifies that the activations at the destination are a copy of the activations at the source in the previous processing cycle.**

Wilson (1993, 1995) postulated that the addition of an extra state-vector to a basic recurrent network like the one shown in *Figure 6.7* should increase its performance. It is considered that, with the extra state-vector, the additional weighted connections back to the hidden layer increase the learning potential of the system. In making comparisons, there is a need to allow for such effects; therefore particular attention was paid to equalising the numbers of weights between the examples of the different architectures tested. Wilson described the design of recurrent networks with different numbers of state vectors, but otherwise similar computational power, and outlined simulation experiments done with such networks. The

results indicated that networks with more state vectors do indeed perform better, although network learning behaviour became erratic with the largest number of state vectors tried (*Figure 6.8*).

The aim of the experiments was to compare recurrent networks having different numbers of state vectors while holding as many other factors as possible constant. The task used was that of predicting the next letter in an English word given an initial string of letters in the word.

The most likely repository of computational information in a neural network is the set of weights. Thus particular attention was paid to equalising the numbers of weights between the examples of the different architectures being tested to avoid the suggestion that hidden units, rather than weights, are a critical resource in a network. This hypothesis was not tested directly in the experiments, but it did turn out that network performance was best for networks with less hidden units and more state vectors.

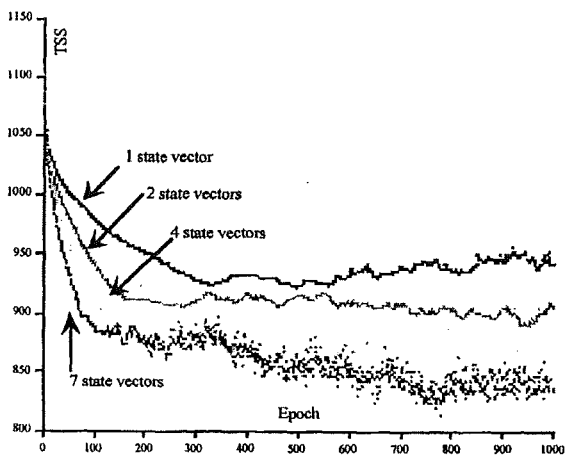


***Figure 6.8* – Typical error plots for a range of multi-state Elman architectures (TSS=Total Sum of Squares).**

Another approach to improving the performance of recurrent networks is the work of Mozer (1992) on induction of temporal structure across longer time intervals by using hidden units that operate with different time constants. This approach is most relevant to tasks that involve recognising the reappearance of a pattern presented in the relatively distant past.

Jordan (1986) and Narendra et al.(1990) studied a class of recurrent networks, sometimes called *sequential* or *Jordan* nets, which use a state vector which contains copies of the *output* layer activations in the previous time step; there are weighted connections from the state vector to the hidden layer (*Figure 6.9*). This differs of course from the Elman network where the state vector is a copy of the *hidden* layer in the previous time step. Both networks can learn sequential structures.

There are a number of other recurrent networks that have been universally recognised. Williams and Zipser designed a network where all hidden and output neural units are connected to all other neural units; if ever there was a network that deserves to be called fully recurrent than this must be it. In contrast Robinson and Fallside (1991) proposed a partially recurrent network which can be considered as a simplified Elman network where only a portion of the hidden layer output acts as a delayed contextual input. They used it as the basis of a speaker-independent word recognition system. The Frasconi-Gori-Soda (FGS) locally recurrent networks (Frasconi et al., 1992; Tsoi et al., 1994) is a multi-layer perceptron augmented with local feedback around each hidden neural unit. The FGS network has also been studied by Mozer (1989).

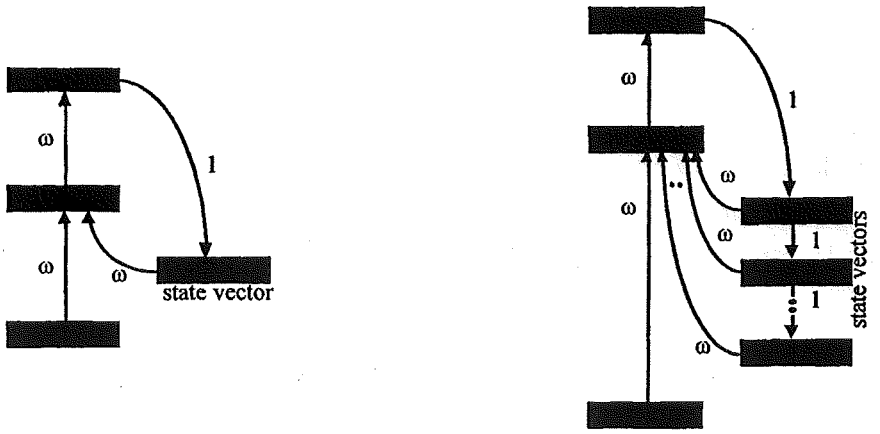


Figure 6.9 - **Architectures of Jordan recurrent networks with single and multi-state vectors: ω signifies trainable weights, 1 signifies that the activations at the destination are a copy of the activations at the source in the previous processing cycle.**

Lawrence et al. (1999) investigated the use of various recurrent neural network architectures (FGS, Elman, William and Zipser) for classifying natural language sentences as grammatical or ungrammatical, thereby exhibiting the same kind of discriminatory power provided by the Principles and Parameters linguistic framework, or Government-and-Binding theory. From best to worst performance, the architectures were Elman, W&Z and FGS. It is not surprising that the Elman network outperformed the FGS network, as the computational power of Elman networks has been shown to be at least Turing equivalent (Siegelmann et al, 1995).

FGS networks have recently been shown to be the most computationally limited (Frasconi et al., 1996). However, Elman networks are just a special case of W&Z networks; as such Lawrence et al. could not explain why the Elman network outperformed the W&Z network, although the suggestion was made that this is a training issue and not a representational issue. Backpropagation-through-time (BPTT) is an iterative algorithm that is not guaranteed to find the global minima of the cost function error surface. The error surface is different for the Elman and W&Z networks, and the results suggested that the error surface of the W&Z network was less suitable for the BPTT training algorithm used. However, all architectures did learn some representation of the grammar. Finally, in similarly veined earlier work Lawrence et al. tested the effect of varying the data input window size. It showed that once again the Elman network, out of all the neural and non-neural methods tested, obtained the best result when used with the smallest temporal input window[6] (Lawrence et al., 1996).

In a *recurrent second-order neural network* the total input to the neuron is not only a linear combination of the components y_j, where each y_j is either an external input or the state of a neuron passed through a sigmoidal function, but also of their *products* $y_j\, y_k$. Moreover, one can pursue along this line and include higher-order interactions represented by triplets $y_j\, y_k\, y_l$, quadruplets, etc. This class of neural networks forms a RHONN and bear a resemblance in their input products to *Functional Link* networks (Pao, 1989).

Kosmatopoulos et al. (1995) have developed learning algorithms for *Recurrent High-Order Neural Network* (RHONN) and analysed their stability properties. High-order networks are expansions of the Hopfield and Cohen-Grossberg (Cohen et al., 1983) models that allow higher-order interactions between neurons. The superior storage capacity of higher-order

[6] Attempts to train networks with small temporal windows failed until several techniques aimed at avoiding local minima were implemented.

neural networks has been shown in Baldi (1988), while the stability properties of these models for fixed weight values have been studied in Dempo et al. (1991). Several authors have demonstrated the feasibility of using these architectures in applications such as grammatical inference (Giles et al., 1991) and target detection (Liou et al., 1991).

6.6 Back-Propagation Through Time: a learning algorithm for Recurrent neural networks

Gradient-descent learning methods form the basis for the training of recurrent networks and encompass algorithms which have been used classically in linear filtering, identification and control, and algorithms which have been established in the framework of neural network research (Nerrand et al., 1994). The variety of algorithms thus available raises the question of a choice of an appropriate one in a given situation.

As mentioned previously, two of the better-known neural algorithms are: BPTT - *Back-Propagation Through Time* and RTRL - *Real-Time Recurrent Learning* (Williams et al., 1989). Other well-known learning methods are *Recurrent Back-Propagation* (Pinenda, 1988) and *Dynamic Back-Propagation* (Narendra, 1991). BPTT and RTRL especially, have a significant number of published variants.

The BPTT algorithm as described by Werbos (1990) can be considered an extension of the back-propagation learning method so that it applies to dynamic systems. It can best be viewed as unfolding a feed-forward neural network in time; growing by one layer for each time-step. The algorithm allows the calculation of the derivatives needed when trying to optimise a recurrent network or, for example in an industrial situation, a control system whose aim could be to maximise some measure of performance summed over time.

If the training data for a recurrent network is divided into epochs, where n_0 is the start and n_1 is the end of an epoch, then it is desired to minimise the square error over the training set:

$$E_{total}(n_0, n_1) = \frac{1}{2} \sum_{n=n_0}^{n_1} \sum_{j \in \Re} \left(t_j(n) - y_j(n)\right)^2 \tag{6.3}$$

this, also known as the cost function, is simply a special case of the well-known method of Least-Squares, used often in statistics. $\Re$ is the set of indices j for those neural units in the network for which a desired target t is specified. The epochwise BPTT algorithm[7] described by Williams et al (1990) allows the calculation of the partial derivatives of the cost function $E_{total}(n_0, n_1)$. This is summarised by Haykin (1994) as follows:

First, a single forward pass of the data through the network for the interval $[n_0, n_1]$ is performed. The complete record of input data, network state (i.e., synaptic weights of the network), and desired responses over this interval is saved.

A single backward pass over this past record is performed to compute the values of the local gradients

$$\delta_j(n) = \frac{\partial E_{total}(n_0, n_1)}{\partial v_j(n)} \tag{6.4}$$

for all $j \in \Re$ and $n_0 < n \leq n_1$ by using the equations

[7] In his classic paper on BPTT, Werbos assumes that the training data forms a single time series, from t=1 to t=T. Thus, in adapting the weights, he always assumes *batch* learning; the weights were always adapted after a complete set of derivatives was calculated, based on a complete pass through all the data. Werbos also starts of with the values of the "memory" weights near zero to avoid local minima and then gradually frees them up.

$$\delta_j(n) = \begin{cases} f'(v_j(n))e_j(n) & \text{if } n = n_1 \\ f'(v_j(n))\left[e_j(n) + \sum_{k \in \Re} w_{kj}\delta(n+1)\right] & \text{if } n_0 \prec n \prec n_1 \end{cases} \quad (6.5)$$

where $f'(\cdot)$ is the derivative of an activation function with respect to it's argument. The use of equation (6.5) is repeated, starting from time n_1 and working back, step by step, to time n_0; the number of steps involved here is equal to the number of time steps contained in the epoch.

(As for *Section* 6.4, the partial derivative $\partial E_{total} / \partial v_j(n)$ can be interpreted as the change in error function E_{total} produced by a change in the internal activation potential v_j of neural unit j at time n.)

Once the computation of back-propagation has been performed back to time $n_0 + 1$, the following adjustment is applied to the synaptic weight w_{ji} of neuron j:

$$\Delta w_{ji} = -\eta \frac{\partial E_{total}(n_0, n_1)}{\partial w_{ji}} \quad (6.6)$$

$$= \eta \sum_{n=n_0+1}^{n_1} \delta_j(n) x_i(n-1) \quad (6.7)$$

where η is the learning rate parameter and $x_i(n-1)$ is the i^{th} input of neural unit j at time $n - 1$.

The computations described here may be viewed as representing the standard back-propagation algorithm applied to a multi-layer feedforward network in which desired responses are specified for neurons in many layers of the network, because the actual output layer is replicated many times when the temporal behaviour of the network is unfolded.

The *epochwise* BPTT algorithm is suitable for off-line operation only and recognising the need for a suitable algorithm that could work on a *continuously* running recurrent network, messrs.Williams and Peng also described a *real-time* version of their BPTT.

Williams et al. (1989) developed their own version of a real-time BPTT algorithm which deviates from the non-real-time version by *estimating* the instantaneous gradient of E_{total} with respect to the weight matrix **W**. This results in an *approximation* to the method of steepest descent. However, this is not a major disadvantage; Haykin (1994) notes that this deviation is analogous to that encountered in the standard back-propagation algorithm used to train the multi-layer perceptron, where weight changes are made after each pattern presentation. While the real-time recurrent learning algorithm is not guaranteed to follow the precise negative gradient of the total error function E_{total} (**W**) with respect to the weight matrix **W**, the practical differences between real-time and non-real-time versions are often slight; in fact the two become almost identical as the learning rate parameter η is reduced (Williams et al., 1989).

There is a another potential problem with the above method, and that is that deviation of the observed gradient descent trajectory from the true gradient following behaviour may itself depend on the weight changes produced from the algorithm, which may be viewed as another source of feedback causing possible *instability* in the network. This inherent instability can be avoided by having the learning parameter η small enough to make the time-scale of the weight changes significantly less than the network operation time-scale (Williams et al., 1989).

6.7 The Multi-step prediction: The Temporal Difference method

In reinforcement learning, considerable attention has been given to the *Temporal Difference Learning* algorithm (Sutton, 1988, 1995, 1996; Cichosz, 1995). This approach focuses on the problem of predicting expected discounted payoff from a given step or state. Temporal difference (TD) methods are a relatively new variation on the neural network scene, which take into account the sequential nature of the problem. Rather than being driven by the difference between predicted and actual outcomes, they are driven by the difference between temporally successive predictions. In this way, learning occurs whenever there is a change in prediction over time. TD learning is a way of extracting information from observations of sequential stochastic processes in order to improve predictions of future outcomes. The TD algorithm was defined by Sutton (1988), and uses the *difference* between such predictions to drive modifications to the parameters that generate them. In fact, Sutton defined a whole class (or superset) of such TD algorithms, known as TD(λ), which look at these differences further and further ahead in time, weighted exponentially less according to their distance by the parameter λ. When $\lambda = 1$, the TD(λ) updating method is identical to the traditional supervised learning approach, (associating all inputs to the final output). On the other end of the TD spectrum where $\lambda = 0$, updating is with respect only to temporally successive predictions.

Sutton illustrated how TD methods can be more intuitive in time series problems with a weather prediction example. The problem is this: A weather forecaster attempts to predict on *each day of the week* whether it will rain on the following Saturday. The conventional approach is to compare each prediction to the actual outcome - whether or not it does rain on Saturday. A temporal difference approach compares each day's prediction with that made on the following day. If a 50% chance of rain for Saturday is predicted on Monday, and a 75% chance is predicted on Tuesday, then a TD method *increases* predictions for days similar to Monday, whereas a conventional method might either increase or decrease them depending on Saturday's actual outcome.

Temporal difference methods have a number of advantages:

- Incremental computation simplifies calculations. The network is updateable every iteration, rather than at the end of the sequence when the final outcome is divulged.
- More efficient use of their experience and faster convergence, leading in some cases to a more accurate prediction.

Temporal difference prediction has been around in various guises for some time. The earliest work in temporal difference methods was due to Samuel (1959) using a checkers playing program. Other proponents of this system included Holland's (1986) bucket brigade, Sutton's (1984) adaptive heuristic critic, and Tesauro's (1992) revolutionary backgammon learning system which has achieved master-level play, surpassing all other commercial backgammon programs (Mair, 1994).

It can be argued that prediction problems can be classed into two main categories; *single-step-ahead* and *multi-step-ahead* prediction. In the latter category the end result is not revealed until n number of steps later, although partial information is revealed at each step. The first category is a special case of the above situation where $n = 1$, or simply where the final result is revealed at every step. The single-step situation is analogous to a supervised learning scheme, where a data pair is presented to a network. Under these circumstances temporal difference methods are equivalent to supervised learning. However, Sutton (1988) argues that with multi-step problems, temporal difference learning provides a significant

advantage over the conventional approach, and that multi-step prediction problems prevail in real world applications. Multi-step problems take full advantage of the TD(λ) paradigm.

Temporal difference learning procedures are expressed as rules for updating the network of weights, w. There are two methods for updating the weights; during the sequence itself, or at the end of the sequence. The latter is known as batch training, and is faster on the whole, but doesn't give the same fast rate of convergence.

For each observation in the sequence, an increment, Δw_t, to the weight vector, w, is calculated, using a formula such as:

$$\Delta w_t = \alpha (z - P_t) \nabla_w P_t \tag{6.8}$$

This is the *generic* supervised learning update formula where z is the (scalar) *outcome* of a sequence of the form $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \ldots, \mathbf{x}_m, z$, where $\mathbf{x}_t$ is a vector of observations available at time t in the sequence. When each $\mathbf{x}_t$ is applied to a network of weights, w, the neural network produces a corresponding prediction P_t, which is an estimate of z. The learning rate, α, is a small number usually around 0.3, which affects the rate of learning. The gradient, $\nabla_w P_t$ is the vector of partial derivatives of P_t with respect to each component of w.

As mentioned earlier, the supervised learning approach is used for correlating observation-outcome pairs. In the case of normal pattern recognition this approach is ideal, since every observation, and every outcome can be different. For a sequential prediction problem this would necessitate using the same z, or outcome value for every observation, because only one outcome is being dealt with. In this situation the normal supervised learning procedure falls down, since all observations lead to the same result, and this value isn't even known until the end of the sequence.

Sutton showed that backpropagation and TD methods are compatible and are able to be combined in an efficient manner with the key being to represent the error $z - P_t$ as a *sum of changes* in prediction, that is, as:

$$z - P_t = \sum_{k=t}^{m} (P_{k+1} - P_k) \qquad \text{where } P_{m+1} \equiv z \tag{6.9}$$

this overcomes the difficulty of not having the ability to update the weights until the final result is known. For intra-sequence weight updating it is important to ensure that both P_t and P_{t+1} are functions of the *identical* set of weights w, although different input vectors. This will mean that changes in predictions are due to changes in the input x, and not on changes in the weights (which could lead to instability).

After substitution into the weight change equation, and suitable manipulation, this becomes:

$$\Delta w_t = \alpha (P_{t+1} - P_t) \sum_{k=1}^{t} \nabla_w P_k \tag{6.10}$$

This equation is equivalent to equation (6.8), but can be computed incrementally, thus allowing increased simplicity in using the equation. It is not necessary to wait until the final outcome is known. The equation (6.10) relies only on the difference of successive predictions, and the sum of all past values of the gradient.

Sutton (1988) formulated a complete class of TD equations, and the above (6.10) is only a special case, which can be referred to as TD(1). The general case is TD(λ):

$$\Delta w_t = \alpha (P_{t+1} - P_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_k \tag{6.11}$$

The new factor, λ, introduced into the above equation is a factor which determines the amount of sensitivity to changes in successive predictions. λ, allows exponential weighting with recency, in which alterations to the predictions of observation vectors occurring k steps in the past are weighted according to λ^k for $0 \leq \lambda \leq 1$.

When $\lambda = 1$, (Widrow-Hoff situation), the estimates of the probabilities for each sequence position are made closer to the final result, much like any supervised learning method. On the other hand, $\lambda = 0$ tries to make the estimate of probability from each sequence position closer to the estimate from the next, without waiting for the final result.

The discounting parameter λ, in TD(λ) determines exponentially the weights of future states based on their temporal difference; interpolating smoothly between $\lambda = 0$, for which only the next state is relevant, and $\lambda = 1$, in which all states are equally weighted (Mair, 1994).

Equation (6.11) can be computed incrementally by breaking down the sum into the following parts:

$$\text{for } t = 1 \qquad \sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_k = \nabla_w P_k \tag{6.12}$$

$$\text{for } t > 1 \qquad \sum_{k=1}^{t+1} \lambda^{t+1-k} \nabla_w P_k = \nabla_w P_{t+1} + \lambda \sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_t \tag{6.13}$$

For $\lambda < 1$, TD(λ) produces different weight changes than normal supervised learning procedures. The difference increases as the value of λ is lowered and is in the extreme when $\lambda = 0$, in which case the weight increment is due only to the most recent observation.

$$\text{for } \lambda = 0 \qquad \Delta w_t = \alpha (P_{t+1} - P_t) \nabla_w P_t \tag{6.14}$$

As Mair (1994) correctly points out; it is possible to see that this equation is structurally very similar to equation (6.8), the supervised learning equation. The difference however relies on the fact that equation (6.14) uses the next prediction, P_t, rather than z. They both use the *same* learning mechanism, but produce *different* errors[8].

Modifications to the basic equation can also allow predictions to be made about events that occur at fixed intervals into the future. This is accomplished with Sutton's (1988) modified TD equation,

$$\Delta w_t = \alpha \left(P_{t+1}^{\delta} - P_t^{\delta}\right) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_k^{\delta} \tag{6.15}$$

where P_t^0 is defined as the actual outcome at time t, and P_t^{δ} is an estimate of the probability of an outcome occurring δ steps later. Simply stated, one makes a prediction for the following step, updates the weights, and carries on repeating this procedure until the required step is reached and the wanted prediction is made.

The *convergence* of TD methods has been examined by Dayan (1992), who concluded that the method of temporal differences will converge to the expected ideal value of prediction with a probability of 1 (Dayan, 1994). Dayan also demonstrated that if the vectors

[8] *Q-learning* is a special form of temporal difference learning where the 'look-ahead' is cut off. Specifically, Q-learning is shown to be equivalent to TD(0) when there exists only one admissible action in each state.

representing the steps are not *linearly independent*, then TD(λ) for $\lambda \neq 1$ converges to a different solution from the least mean squares algorithm of Widrow-Hoff.

Cichosz (1995) examined the issues of the efficient and general implementation of TD(λ) for arbitrary λ, for use with reinforcement learning algorithms optimising the discounted sum of rewards. The traditional approach is argued to suffer from both inefficiency and lack of generality; the TTD (Truncated Temporal Differences) procedure is proposed as an alternative, that indeed only approximates TD(λ), but requires less computation per action and can be used with arbitrary function representation methods. The idea from which it is derived is fairly simple and not new, but probably unexplored so far (Cichosz, 1995). Encouraging experimental results are presented, suggesting that using $\lambda > 0$ with the TTD procedure allows one to obtain a significant learning speedup at essentially the same cost as usual TD(0) learning.

When judged by the preponderance of available reviews and papers it appears that the TD algorithm's ability for learning sequential decision rules, and thus strategy, is a major reason for its utilisation. An interesting facet of these applications is in the "intelligent" (AI) game, which in general stresses the ability of TD to learn a particular game evaluation function, e.g. Altman et al., 1993; Allis, 1994; Mair, 1994; Harmon et al., 1995; Rossin et al, 1996. TD learning is very attractive for board games, because (i) the TD method determines the eligibility of each board position for receiving a credit from the game outcome, and (ii) the neural network predicting the evaluation function can learn by playing against itself.

A very recent example of this is given by Zaman et al. (1999) whom examine a game known as Go. Go is a two-player perfect information board game of strategy played on a square 19x19 grid with unlimited supply of two types of game pieces, e.g., black stones and white stones, respectively, for the two players. The objective for each player is to surround as much empty grid points as possible with his stones. The conventional tree search algorithm for computer games fails to build a good Go program, mainly because it has a computationally intractable board evaluation function. Expert players evaluate board positions by using their skills at pattern recognition and these are very hard to capture in algorithms. Zaman et al.'s paper uses a committee of neural networks and temporal difference methods to learn the board evaluation function for 9x9 version of Go (this smaller board is used by beginners).

6.7.1 Temporal Difference networks and FEMS

It can be concluded that use of the TD method for FEMS is only plausible if there is a need for multi-step prediction. The conventional prediction learning methods utilise an error function based on the difference between predicted and actual outcomes; in TD methodology it is the difference between temporally successive predictions. Fluid demand, such as hot water utilisation in a domestic household, could be viewed as a multi-step prediction problem where partial information about the correctness or incorrectness of the forecast is gradually revealed (during say, a 24 hour period) in a series of time-steps subsequent to the initial prediction having been made. In single step prediction problems, all information about the correctness or incorrectness of each prediction is revealed at once. The temporal difference method can take its advantage from multi-step prediction problem because learning occurs whenever there is a change in prediction over time.

So, using the previous 'weather' example as a template, if at midnight on Sunday a total (24 hour period) hot water energy consumption of 150 litres (in equivalent kJ) is predicted for Monday midnight; and 3 hours later, at 3 o'clock on Monday, a 100 litres total for Monday midnight is predicted, then the temporal difference method should decrease subsequent

predictions for times with conditions similar to 3 o'clock. This would be repeated for every 3-hour period up until midnight. For a feed-forward network trained with this method the predictions should be more accurate than a network trained with, for example, back-propagation, because the final outcome is more closely related to the later time-periods in the sequence than it is to the earlier periods. However, all this comes at the cost of additional calculations and shows no immediate advantage c.f. a single prediction made with a dynamic temporal network such as TDNN or RNN.

6.8 Discussion

The issue of time prediction has been examined in some detail in this chapter. Having arrived near to the end, it is an appropriate point at which to select the most suitable network and summarise its operation, as well as the potential, for the Fluid Energy Management system (FEMS).

The *single* or *multi-state recurrent Elman architecture* was chosen as the most potential candidate for the FEMS because (i) of its reportedly successful application in a significant number of cases, (ii) its simplicity, (iii) proven ability to dynamically process temporal data and (iv) it has been shown that recurrent Elman networks are able to significantly reduce the noise level in process measurements without explicit knowledge of the non-linear dynamics of the system (Karjala, 1992).

Many researchers using neural networks for time series prediction have chosen to "parallelise time" by incorporating a moving window of input values for *standard* feed-forward networks. The past N values of the M input variables might be used simultaneously as the network inputs resulting in networks with N x M input nodes. The length of the data window N must be long enough to capture the dynamics of each variable, but the number of nodes must be kept to a minimum in order to minimise the size of the network. In general, the size of the data window must be determined by trial and error, and each input variable in *multivariate* time series should have a separate data window size for optimal performance. This adds to the number of parameters that must be tuned by trial and error when using, for example, the standard back-propagation training technique.

The use of a recurrent neural network instead of an MLP with a window of time delayed inputs has the ability to overcome this unwanted increase in input parameters by explicitly addressing the temporal relationship of the inputs via the maintenance of an internal state. This should help to make the problem less ill-posed.

Recurrent networks include links between nodes that feedback signals to other nodes on the same or prior layers. This provides networks with internal states and a form of memory. As a result, recurrent networks provide more than the simple 1-to-1 mapping of feed-forward networks. The output of such networks depends not only on the current inputs but also on previous inputs. Time is represented implicitly rather than explicitly through the use of a moving window.

The use of a recurrent neural network is important from the viewpoint of the curse of dimensionality because the RNN can take into account greater history of the input. Trying to take into account a greater history with an MLP, by increasing the number of delayed inputs, results in an increase in the input dimension. This is undesirable, given the fact that a black box FEMS should work with an as small a neural network configuration as possible.

(The *curse of dimensionality* refers to the exponential growth of hypervolume as a function of dimensionality. Consider $\mathbf{x}_i \in \Re^n$. The regression $y = f(\mathbf{x})$, is a hypersurface in $\Re^n$. If $f(\mathbf{x})$ is arbitrarily complex and unknown then dense samples are required to approximate the

function accurately. However, it is hard to obtain dense samples in high dimensions [9]. This is the "curse of dimensionality". The relationship between the sampling density and the number of points required is $\approx N^{1/n}$ where n is the dimensionality of the input space and N is the number of points. Thus, if N_1 is the number of points for a given sampling density in 1 dimension, then in order to keep the same density as the dimensionality is increased, the number of points must increase according to N_1^n (Lawrence et al.,1997)).

RNN architecture is similar to the standard feed-forward architecture with layers of input units, hidden units, and output units, but also includes a set of context units which save the prior activation of the hidden units and feedback the stored activation of the previous cycle to the hidden units in a fully connected manner. For the application under consideration, the input vectors of historic data could correspond to either a (multi-variate) single step or a window of past time steps if so desired. The target vector, for supervised training, would be the data series value(s) at the next time step. Due to retention of past information in the recurrent connections it should suffice to present very few past samples at the input. This is of great advantage in reducing the amount of storage memory needed for a black box energy management system.

The number of hidden nodes is equal to the number of context nodes and will be adjusted to fit the problem at hand. The input layer and the hidden layer should have one bias node each, and *tanh* or *sigmoid* activation functions are intended to be used throughout the hidden layers of the networks.

Determining the optimal architecture is not trivial. For example, if long-term memory involves a function of the inputs, rather than the inputs themselves, one or more non-recurrent layers should probably *precede* a recurrent layer. Similarly, if the output is a non-linear function of the long-term memory, a non-recurrent layer probably should *follow* the recurrent layer. It is possible to avoid such considerations by designating all the layers to be recurrent, but at the cost of many additional parameters with the associated computational penalty.

Most recurrent and ordinary feed-forward networks used today are trained using some form of back-propagation. The goal is to minimise the squared error between the actual network outputs and the target values by adjusting the weights in the network for all the output nodes and patterns. An alternative algorithm that exists today for unconstrained optimisation is the *BFGS quasi-Newton algorithm*. For a detailed theoretical discussion see Luenberger (1989). This method has outperformed other methods on comparably sized problems (Karjala et al., 1992) but works best on small problems, with less than approximately 100 weights.

6.8.1 Data input

The problem of learning from examples is fundamentally ill-posed, i.e. there are infinitely many models which fit the training data well, but few of these generalise well. In order to form a more accurate model, it is desirable to use as large a training set as possible. However, for the case of highly non-stationary data, increasing the size of the training set results in more data with statistics that are less relevant to the task at hand being used in the creation of the model.

9 *Kolmogorov's theorem* shows that any continuous function of dimensions can be completely characterised by a one-dimensional continuous function. In other words, for any continuous function of arguments, there is a one-dimensional continuous function that completely characterises the original function. As such, it can be seen that the problem is not so much the dimensionality, but the complexity of the function, i.e. the curse of dimensionality essentially says that in high dimensions, as fewer data points are available, the target function has to be simpler in order to learn it accurately from the given data.

When the Fluid Energy Management System has accumulated an initial minimum amount of historical data (a quantity to be determined by test trials), the recurrent network will have input vectors consisting of parameters such as yesterday's value only or 2 weeks of historic energy usage, the time of day (*only* if *multiple* predictions are made for each day), day of the week, month of the year, whether it's a regular working day or a holiday, ambient temperature, and possibly humidity. It is prudent to input the additional parameters as they are deemed to have an effect on the usage of hot water in the household in the form of additional hot showers/baths, attributable to perspiration on hot sunny days, sport activities, school and working day as opposed to weekends, etc. For this specific case the neural network should extract, during batch (offline) training, the relevant patterns and associate them with different hot water energy use for different days of the week and times of the month/year. One characteristic of this problem is that the hot water usage for a given day depends only on that day's ambient condition. However, when looked at, say, for every 3-hour period the usage is determined not only by that period but also on the trend represented by the conditions from the few hours before.

Chapter 7. Energy Management Systems

7.1 Introduction

Why manage energy? – Because it makes good economic and organisational sense. A well designed, well-maintained and well-operated facility, industrial or otherwise, will be energy-efficient and offer a higher degree of amenity to its users. For commercial and government organisations managing the consumption of energy is an important element in the process of providing cost effective services, and minimising the indirect cost is passed on to the community. Energy audits for large office buildings indicate that more than 20% of energy consumed can be saved by implementing measures which will pay back the investment in less than three years. Management, operating and maintenance measures that require little or no capital investment can achieve a 10% saving in energy usage. These energy savings translate to commensurate cost savings and reductions in greenhouse gas emissions.

Greenhouse gas emissions arising, for instance, from the combustion of fossil fuel for thermal power generation, are now universally accepted as having profound and detrimental effects on our global climate. New Zealand is signatory to the *Kyoto Protocol*, an international agreement prepared in December 1997 under the *United Nations Framework Convention on Climate Change*. Under the Kyoto protocol, New Zealand is permitted to keep its greenhouse emissions at the present level, which represents an overall reduction on 15-year projection estimates.

Energy management is a program of well planned actions aimed at reducing an organisation's energy bills while offering improvements in comfort for users and reducing detrimental environmental impacts.

Energy management involves:

- devolving responsibility for energy bills to those with the authority to change the way energy is used;
- providing resources where required;
- collecting and analysing existing energy use data;
- undertaking an energy audit to determine where, and how efficiently, energy is used;
- implementing energy saving measures;
- regularly reporting the savings that have been achieved.

There are two central energy management strategies:

Energy conservation - the avoidance of wasteful energy use and the reduction in demand for energy-related services (e.g. if you don't need it - turn it off).

Energy efficiency - the reduction in consumption of energy for current operations (e.g. if you need it - do it more efficiency).

Appropriately applied energy management strategies will lead to a significant reduction in the costs for large corporations and government services, and improve the quality of services provided. For the domestic user the benefits of energy efficiency go beyond simple dollar savings. Lower utility bills result in increased disposable income for homeowners and profits for businesses. Some of this money will be spent in the community, providing local economic development and jobs.

7.2 Energy utilisation in New Zealand

It is interesting to briefly examine the generation and consumption of electricity as well as the overall energy utilisation picture for New Zealand. This will give some indication of the extent of the benefits to be reaped by the proposed FEMS when heating only that quantity of hot water, or other fluid medium, that is demanded (it is more difficult to judge what savings would be achieved by simultaneously lowering the peak-demand curve).

In New Zealand, *hydroelectricity* provides just over 70% of the electricity supply. Except during infrequent prolonged dry periods when shortages have occurred, many people generally view electricity as plentiful, derived from clean sources and possibly even running to waste if not used. However, even in New Zealand a significant proportion of electricity generation is derived from *thermal* stations using fossil fuels, mostly natural gas, and in a normal year electricity consumption would have to be reduced by 20% before significant spilling from hydro lakes occurred (CAE, 1996).

Table 7.1 shows the 1995 figures for New Zealand energy supply. It shows that natural gas is the most important so-called *primary* energy resource to New Zealand, that is the production of raw energy before conversion and losses. Only a fraction of the supply, about 20%, is used directly as a reticulated gas supply to homes, offices and industry. The balance is used in energy transformations. Hydroelectricity only comprises 14% of the primary energy supply. Hydro resources make up 19% of the final end-use or consumer energy supplies. In round terms, about 70 PJ[10] is used to produce electricity and, until around 1993, 60 PJ was used to produce synthetic petrol. Now, production of synthetic petrol has largely ceased in favour of producing more profitable chemical grade methanol.

After natural gas, liquid fuels are the next most important primary energy source and they represent the main form of consumer energy. The conversion of natural gas to synthetic petrol is one of the reasons for the liquid fuels versus consumer energy figure being greater than the primary energy figure.

Energy Source	Primary Energy	%	Consumer Energy	%
Natural Gas	180 PJ	30%	40 PJ	10%
Liquid Fuels	165 PJ	28%	170 PJ	43%
Geothermal	90 PJ	15%	20 PJ	5%
Hydro Power	80 PJ	14%	75 PJ	19%
Other Electric	NA	-	25 PJ	6%
Coal - Bit/Lig.	50 PJ	8 %	45 PJ	11%
Wood/Pulp Waste	25 PJ	4%	25 PJ	6%
TOTALS	590 PJ	99%	400 PJ	100%

***Table 7.1* - Indicative New Zealand energy supply, 1990-95** *(Collins, 1995).*

Note: Geothermal energy is used to produce approximately 7 PJ of electricity and 13 PJ of heat.

The amount of geothermal energy tapped as a primary energy source gives a misleading picture of its importance. The conversion of geothermal energy to consumer energy,

[10] PJ = Peta Joules, 10^{15} Joules

especially the transformation to electricity, is very inefficient. Added together, the heat and power derived from geothermal energy makes up about 5% of the consumer energy total.

The overall picture is one of heavy reliance on non-renewable fossil fuel energy resources, and natural gas and condensates in particular. Natural gas is a valuable energy resource. It is relatively clean burning and readily controlled. It can be efficiently used for space, water and industrial process heating. It creates the least amount of greenhouse gas emissions of all fossil fuels. The efficient use of natural gas and the electricity and synthetic petrol derived from it is strategically important. Careful use of electricity from hydro resources also reduces the need for natural gas thermal generation.

Table 7.1 also shows the large losses incurred in transforming primary energy to consumer energy, although the ratio is skewed by the inefficient conversion of geothermal energy. If this source is set aside, then the overall ratio of energy conversion and transport/transmission to consumer energy is 75%. The major sources of losses are in thermal electricity generation and synthetic petrol production, where the ratios are less than 50% (CAE, 1996).

After liquid fuels, electricity generation is the energy supply likely to experience the fastest growth. Power demand is expected to grow from around 30,000 GWh (108 PJ) at present to around 45,000 GWh (162 PJ) or more by 2020, an annual rate of increase of 1.6%. The extra load will be met by constructing a variety of new power stations hydro, geothermal, windfarms, natural gas combined cycle, co-generation plants and coal-fired plant. By the year 2010, an extra 1500 MW may need to be built and by the year 2020, an extra 2300 MW of capacity may be needed. The image of New Zealand as a country deriving its electricity from renewable sources will not be improved:

- hydro-based share falls from 70% in the 1990s to around 55%, despite the construction of new hydro stations;
- geothermal's capacity increases from 5 % to 11 % by 2020, but not enough to compensate for the reduced hydro share;
- wind farming becomes established, but by the year 2020 still only contributes 2%; and
- the share of coal rises from almost nil in 2000 to 18% by the year 2020.

It is without a doubt that the need to build extra power stations will eventually cause a substantial increase in the price of electricity; especially in a "user pays" scenario. Prices for residential consumers in the year 2020 could exceed 30 c/kwh as a result (a 100% increase over 1999 levels). Although renewable energy sources (such as hydro, geothermal and biomass) as a proportion of total primary energy may grow slightly from 1990 to 2020, their share of consumer energy is expected to drop from 30% to 23%. This is due to the fact that while geothermal primary energy use doubles, its poor conversion efficiency means it has little impact on consumer energy supplies.

As stated earlier, hydro provides just over 70% of the electricity supply, being 75 PJ, with the country as a whole consuming 30,000 GWh (108 PJ). In 1999 terms this means that approximately 110 PJ of electrical energy is used by the domestic, commercial and industrial consumers combined.

The domestic sector has a large impact on this country's electricity consumption with an overall figure of 37%. This makes it worthwhile to invest in energy efficient technologies, of which FEMS would be but one example. The threat of future electricity shortages would reduce, and the need for additional power generation facilities could be delayed. An indication of the huge potential and important role private homeowners and occupants have

and should further develop in saving energy can be seen by the 1991 figures released by the Ministry of Commerce, which claim that current domestic total energy consumption can be reduced by 60%! At present, the average homeowner spends a significant proportion of the household budget on buying energy, whether it be electricity, gas or solid fuel. *Figure 1.1* from *Chapter* 1 illustrated that around 65% of the household energy bill goes into space heating and providing hot water, and these two areas are where considerable energy savings can be made. At present, approximately 84% of cooking appliances and 87% of hot water cylinders use electricity.

As stated earlier, the domestic sector accounts for 37% of the *total* electricity consumed in New Zealand, making it the second largest user behind the industrial sector at 42%. This results in an energy consumption figure of almost 11,110 GWh (40 PJ) for the domestic side. Recent research by Stoecklein et al. (1998) on domestic energy consumption in two typical New Zealand cities (Figure 7.1) confirms that heating water is still a dominating factor in electricity usage. *Figure 1.1* showed that the typical household uses 45% of the energy supplied for water heating, being equivalent to 5,000 GWh (18 PJ). This means that, potentially, a successful fluid energy management system used by, for instance, half the domestic households in NZ could save an estimated 10% on water heating and this would result in an annual energy saving of 250 GWh (0.9 PJ). If in addition a good proportion of the industrial sector also embraced the FEMS than the energy savings could be even more substantial, possibly negating for a number of years the forecasted annual energy rate increase of 1.6%.

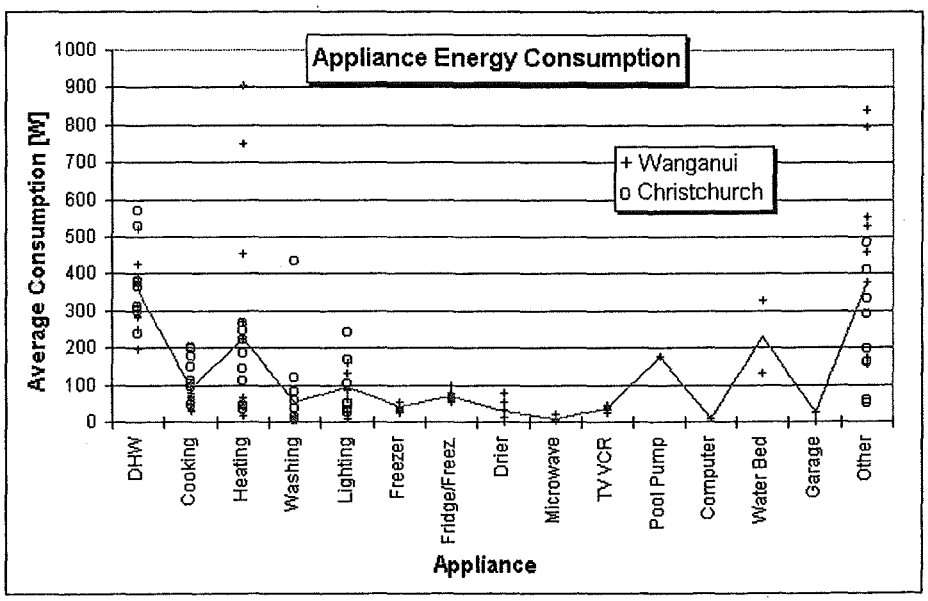


Figure 7.1* – Appliance energy consumption in a North Island city (Wanganui) and a South Island city (Christchurch), *(Stoecklein, 1998).

If energy efficiency is to become a priority in the domestic sector, the cost implications must be beneficial in the long run, and people must realise the savings potential and act on it. Any extra costs for energy efficient apparatus and other measures should quickly be repaid from energy cost savings.

CAE (1996) notes that the relatively low price of power in this country discourages energy efficiency, although at today's prices, there are many cost-effective energy efficiency options.

A good example of this is provided by the upgrading in the New Zealand Standard NZS4606:1989 for thermal insulation around new hot water cylinders. It has been estimated (EECA, 1991) that a 270 litres cylinder fitted with a high grade of insulation could save 700 kW of electricity annually; equating to around $100.- with current prices. A retrofit solution for older cylinders is available and costs less than this. While the differences may be reduced where a household has frequent and heavy hot water draw-off, it is still substantial and justifies the extra expenditure.

If the electricity rates were higher, the range of potential efficiency options would expand. While increased power prices would draw household attention to energy efficiency opportunities, it introduces a problem. Many households, mainly those on low incomes, may only be able to respond to hire prices through behaviour changes and conservation rather than investment in efficiency hardware. These people may have great difficulty finding the capital for such investments as FEMS, even though a home economics study may show that this would improve the family finances. Access to reasonably priced finance is a major issue. One possibility here is for the local power company to fund certain proven investments and recover the capital and interest via power bills.

Many low-income families are in rental accommodation and this exacerbates the already unequal access to energy efficiency. It may not be in the family's interest to fund non-chattels, that is improvements that cannot be taken when the family moves. The landlord has no incentive to invest in house improvements as the resident family pays the power bills. This situation is a case of market failure. Solutions include building codes that require upgrading of old residential buildings, sources of capital for low income families, and community action to encourage people to group together and share resources and skills.

7.3 Energy Efficiency

Energy efficiency means using less energy to get the same result, or getting more results from a given amount of energy. The question of the potential for energy efficiency in New Zealand can be addressed via an industrial/domestic sector and technology approach.

With a sector- and technology-based approach, *cost-effective technologies* and *management-systems* applicable to a sector are identified. The opportunities for implementing these technologies across the sector and the degree to which these have already been adopted are assessed. This approach provides an indication of the amount of energy efficiency improvements that remain under present economic conditions.

Efficiency gains might come about through better use of a single energy source, electricity say, or through fuel switching. Fuel switching can work two ways. Replacing electric resistance loads with natural gas use can improve efficiency. Replacing natural gas, coal and oil plant with electro-technologies (using mechanical vapour recompression to dry timber, for example) can also improve energy efficiency. As noted earlier, efficiency improvements can also be made on the supply side, such as through better conversion of primary energy. The following outlines recent estimates of the energy efficiency potential that is likely to be available in New Zealand.

In 1992, the Electricity Corporation (ECNZ, 1992) released a report entitled "The Developing Market for Energy Efficiency in New Zealand". This paper developed and analysed a number of scenarios, each with different assumptions about the introduction of energy efficiency technologies between 1990 and 2005. This work provides an idea of the maximum energy efficiency potential and the implications of trying to achieve this. A modelling exercise was used to examine two scenarios, the standard and efficiency scenarios.

The *standard* scenario represents a business-as-usual approach to energy efficiency and energy use. Energy efficiency is taken as a low business priority. The scenario uses existing rates of turnover for equipment and building stock. Existing technology is assumed to he replaced in most sectors by the currently best available technology within 15 years. The currently best available technology means equipment and practices that are both commercially viable and considered the most efficient means of providing the energy service in question.

The *efficiency* scenario uses accelerated rates of equipment turnover and assumes that state-of-the-art technology would progressively replace existing technology in most sectors within 15 years. State-of-the-art technologies are the most efficient means of providing an energy service using current knowledge. Compared with the best available technology, they use less energy and provide lower operating costs, but are usually more expensive to install and have longer pay back periods. The *FEMS systems* of *Chapters* 8 and 9 are typical examples of such technology.

The key points of comparison between the two scenarios are presented in *Table 7.2.*

Technology improvements and increased use of co-generation under the efficiency scenario help to break the GDP-energy use relationships experienced up to 1990. Consequently, there is a 47% increase in GDP with an 8% reduction in consumer energy. The standard scenario has a 45% growth in GDP, but a 13% rise in consumer energy.

Parameter 1990-2005	Standard	Efficiency
Primary Energy	plus 14%	fail 2%
Consumer Energy	**plus 13%**	**fall 8%**
GDP Growth	plus 45%	plus 47%
Additional Generation	plus 500 MW	Nil
Carbon Dioxide Emissions	plus 36%	fall 4%
Primary Conversion Eff.	down 1 %	gain 3%
Consumer Conversion Eff.	gain 1%	gain 11%

Table 7.2 **-Energy efficiency scenarios - key results,** *(ECNZ, 1992).*

The extra investment required for the efficiency scenario pays off well in the medium and long term. Using state-of-the-art technology does not hinder economic growth. Instead, it lowers energy costs and provides significant environmental returns. It would neutralise electricity load growth until at least 2005 (the end of the scenario analysis period) so that no new power stations would be needed in the interim. The resultant improvement in all forms of energy use translates into a 4% reduction in carbon dioxide emissions from 1990 levels by 2005.

The efficiency scenario involved investments that some businesses may not be able or willing to make at present. The standard scenario involved many lost opportunities in that not all businesses were assumed to invest in energy efficiency even though this would be profitable. The true potential for cost-effective energy efficiency in New Zealand probably lies between the two scenarios.

The Energy Authority states that a 1993 report to government officials provided a conservative estimate of actual energy efficiency potential (Harris, 1993). The report did not take a best case approach, that is, it did not assume that all cost-effective opportunities would be taken up. Instead, an attempt was made to identify realistic penetration rates for each technology given a concerted government funded program. The program proposed the adoption of minimum energy performance standards, the development of a market for energy service companies and the implementation of a comprehensive communication strategy to reduce domestic energy use.

Ministry of Commerce 1992 forecasts were used to produce a baseline energy use pattern for the year 2005 and energy efficiency opportunities were measured against-this baseline. Energy use in the *domestic sector* had the potential to be 11 % lower than the baseline in 2005. For this sector, energy use could in fact fall by almost 5% between 1990 and 2005. Energy use in other sectors would rise even with a concerted effort, but considerable savings were possible compared with the business-as-usual baseline case.

The commercial sector (retailing, services, institutions, etc.) had the potential for a 8.5% savings through energy efficiency, compared to the business-as-usual case, by 2005. It was assumed that energy efficiency investments in the major energy using industries - aluminium, iron, steel, forestry and petrochemicals would take place without a special programme and would therefore form part of the baseline forecasts. A potential saving of 8% from investment in other industries was identified. The potential in the transport sector was thought to be 3%. The transport savings would largely occur as a result of commercial competition and through vehicle manufacturer initiatives.

The uptake of energy efficiency improvements in New Zealand has not been high, notwithstanding the incentives provided by the benefits outlined above. This suggests a degree of failure of government policy and market performance in the past. Energy markets in New Zealand have been reformed and a major issue is whether they will do a better job of promoting energy efficiency than the previous arrangements. In the electricity sector, for example, power generation, distribution and energy supply have been unbundled and are now operated as distinct businesses. Power companies have had to choose whether they want to operate a line business that takes care of the local supply network or an energy supply business that focuses on selling electricity.

A study undertaken for EECA in 1994 indicated that the sum of all cost-effective opportunities could be more than twice the estimates provided in the Ministry of Commerce Report (EECA, 1994). The challenge appears to be to find ways to motivate people to explore their energy efficient options and act on them. An important point is that once all the current cost-effective opportunities are taken up, that is not the end of the matter. Rising energy prices and/or innovation in efficiency technologies that lower costs or increase service open up a new range of opportunities.

7.4 Conserving energy and increasing efficiency for the hot water cylinder

Before going into the detail of the FEMS it is worthwhile to examine/recap the potential energy savings that are possible not only by installing a good energy management platform, but also with proper insulation, additional components and optimum parameters settings for the typical domestic hot water cylinder. FEMS, when viewed as an after-market or retro-fit system, can be considered as just such an 'additional component' and, as described in *Chapter* 2, needs to be able to adapt itself to the large variation in household circumstances it will encounter. As such, it must be seen as an adaptive means of complementing the other energy saving methods.

The functioning of an efficient hot water system can be illustrated by considering cylinder size, temperature setting, the use of tempering valves and a common end use, such as showering. Except for lukewarm clothes washing, most household use requires a mix of hot and cold water, resulting in a temperature of at least 40^0C. The capacity of a system can be taken as the amount of water it can provide above 40^0C, after mixing the hot water with ambient cold water.

In many New Zealand houses the thermostats of the hot water cylinders are set to 65^0C or higher, whereas a temperature at the tap of around 50^0C to 55^0C is safe and meets the maximum temperature requirements for household use (ACC, 1990). Lowering the cylinder temperature from 65^0C to 55^0C will reduce the cylinder standing heat loss by 20%, but will also reduce the effective capacity by 40%.

Many water cylinders are undersized in that they could not meet household requirements if set to 55^0C. Maximum system demand (litres per hour) is typically *twice* the average demand. Factors such as increased numbers in the home, successive loads of washing and lower incoming water temperatures (in winter) all impact on maximum demands. Houses change hands and family sizes vary. The most common way to accommodate these factors is to adjust the cylinder thermostat (CAE, 1996); this lends credence to the chosen method of cylinder energy control in FEMS.

When cylinder insulation-effectiveness and water-use patterns are considered, lowering the temperature to simply reduce standing heat-loss may seem hardly worthwhile. Expressed as a percentage of total hot water consumption, the heat loss of an well-insulated cylinder is theoretically between 12.5% and 15% of total consumption. Therefore, reducing the temperature from 65^0C to 55^0C will, theoretically, achieve a 3% improvement in *overall* household energy consumption (CAE, 1996). This seems a reasonable assumption and is confirmed to some extent by the results from the trials undertaken for this thesis which have shown a reduced heat loss of ±20% of total consumption over a 24 hour period for a fully heated system (no withdrawals made). However, the CAE figures ignore the daily substantial saving attained in not having to heat 180 to 270 litres of water an additional 10^0C, a process which takes roughly 40 minutes in the 180 litres capacity test cylinder of *Chapter* 2 and, by extrapolation, a good hour for a 270 litres hot water cylinder. Fitted with a 3kW heating element and at a price-charge rate of NZ$0.13/kWh, this could save the household approximately NZ$150.- in water heating bills on an annual basis.

Nevertheless, in terms of retaining the energy that has been put in, a well-insulated cylinder provides a superior efficiency improvement c.f. lowering the water temperature. The cylinder should always be of a high insulation grade or it should be wrapped in extra insulation. Other main factors for an efficient system are:

Insulating all pipes and valves that carry hot water.

- Fitting the cylinder with a consumer-adjustable thermostat so that it is possible to operate it at the lowest possible temperature commensurate with a satisfactory performance (i.e. adequate capacity).
- Fitting a tempering valve to maintain delivery temperature below the temperature of the water stored in the cylinder.
- Utilising efficient end-use equipment and appliances.
- Using an efficient cylinder venting system.

A tempering valve (installed at the cylinder outlet) working at 50^0C will reduce pipe losses by up to 25% and reduce hot water consumption by appliances with coarse temperature selection, such as washing machines, by 15%. It also promotes overall system-safety by eliminating the potential for scalding.

The main use of hot water in most households is for showers. Many showerheads put out 12 to 16 litres per minute. An efficient showerhead may only use 7 to 8 litres. Some people may experience a slight loss of amenity at this flow rate. In such cases, even if the flow rate was cut back to only 10 litres per minute, a hot water saving for this use alone could be 25% to 30% (CAE, 1996).

Hot water cylinders require a means to relieve temperature and pressure build-up due to water expansion, failure of the thermostat etc. The attachments used to achieve this can be sources of energy loss. *Low-pressure* systems usually have an *open vent pipe* from the top of the cylinder. This typically discharges onto the house roof, although in some cases the discharge can be back into the header tank, where one exists. Neither approach will be energy efficient unless the whole vent pipe is well insulated. This is because even when the vent is not discharging, thermal circulation currents will exist in the pipe, taking heat from the cylinder and radiating it from the vent pipe.

An alternative to insulating the whole vent pipe is to fit a *heat trap*. A heat trap is a full loop or downward U bend in the vent pipe that cuts off the thermal circulation. Water above the trap will remain cold. The vent pipe leading up to the heat trap should be insulated Some low-pressure systems do not use an open vent pipe, but have a *pressure relief valve* instead. If this relief valve is at the top of a tall vent pipe, then again good insulation is essential. The best place for the relief valve is right beside the cylinder. Converting open-vented systems to relief valve systems improves energy efficiency and increases water pressure.

Mains- or high-pressure systems have a *temperature* and *pressure relief valve* at the top of the cylinder. As water is heated it expands and it is possible to vent around 3% to 4% of the volume of the water heated. A *cold-water expansion valve* attached to the cylinder *supply* line avoids this loss. The use of line filters is recommended. By filtering out grit these reduce the risk of valves and taps not closing properly and reduce tap seat wear. Dripping taps and valves can be important causes of energy loss. More information on improving the energy efficiency and performance of hot water systems is available in a booklet from EECA (EECA, 1995).

One of the better examples of the few available (although not in NZ) commercial controllers for domestic hot water cylinders is shown in *Figure 7.3* and is produced by Protemp in the United States. It claims the following:

> "PRO-TEMP's energy efficient controllers begin saving you money on the first day of operation, automatically matching hot water temperature to demand. The controller continuously monitors and memorizes daily hot water usage patterns (as shown in *Figure 7.2*) and tells your water heater/boiler precisely when to raise and lower temperatures. Most models have modem capability and can provide seven days of detailed graphs and trend graphs for the last 12 months. With PRO-TEMP online, you can save up to 25% or more of your fuel costs. Depending on the model selected, our Water Heater Controllers (for large facilities) offer other features such as Time Scheduling, Multiple Staging of Water Heaters, Automatic Faxing of Graphs, Security Password. PRO-TEMP controllers are designed to operate with natural gas, propane, steam, oil, and

electricity. There is no need to cut pipes or interrupt service. Hot water is always available to building occupants during installation."

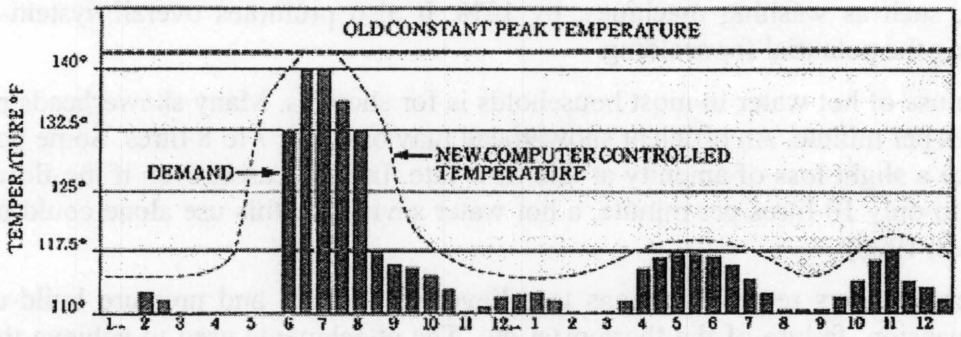


***Figure 7.2* – The claimed operation of the Pro-temp energy controller.**

Pro-temp smallest controller, the HD-60, is suitable for domestic household installation; Pro-temp has this to say about it:

"The HD60 Series hydronic controller delivers impressive savings to smaller properties with one or two boilers. This control is easy to install and operate. Programming is simple and completed from a pressure-sensitive keypad on the front panel. The easy-to-read 4 line LCD screen with display supply, desired and outside temperature and heater demand at 10 minute averages for the last 7 days (HD64 only).

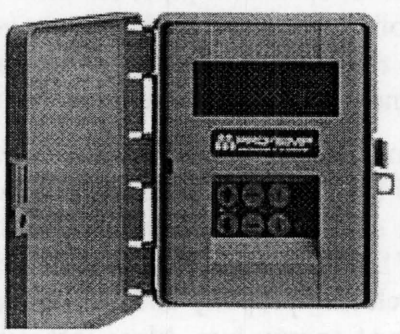


***Figure 7.3* – The HD-60 series hydronic controller suitable for domestic installation.**

This hydronic series offers features usually found in more costly controls. And depending on the model selected the HD60 series offers features such as Time Scheduling which provides up to 20 different time periods per week. For those times when extra heat is needed, a Bypass Timer is provided. When the controller is set in this mode, it will run for the time programmed, then return automatically to the run mode. The Domestic Hot Water Control Feature keeps the boiler running at a reduced temperature for supplying domestic hot water during the summer months. The pump will turn off, but the control will maintain this temperature. *Want to know how much you're saving?* The automatic savings test will let you know by providing a screen which shows the percent of energy savings. *Payback is typically less than two heating seasons!*"

Obviously this system has no form of artificial intelligence and appears to simply collect time stamped data at either 10-minute or half-hourly intervals over the period of one week, which is stored in memory and acted upon each day at around the same time. It is safe to assume that it has some form of averaging mechanism for data collected at the same time interval each day. It is questionable whether the HD-60 even has automated data-collecting

feature; it is more probable that the consumer has to input his own heating schedules (up to 20 different time schedules per week). It also assumes that the consumer is happy to heat water at any time of the day, and in fact in the U.S. this might not be much of a problem as most houses have gas or fuel-oil as their means of energy input. The Pro-temp controller appears to be suited to this segment of the domestic market.

Presented as shown it would appear safe to state that the FEMS is a far more capable system than the Pro-temp HD60 and offers greater potential than Pro-temp's more intelligent controllers which are aimed at hotels and large buildings.

7.5 Energy Management Systems

7.5.1 EMS development

Energy Management Systems (EMS) have made a substantial impact on the control of buildings and manufacturing services plants and on energy efficiency. Their development has been extremely rapid; there were few systems in 1980, whereas most new commercial, industrial and public buildings in the 1990s have some form of a EMS. Perhaps because of this rapid development, there is still a lot of unused potential in many current systems.

Energy management systems have developed alongside, and been a result of, the microelectronics and computing revolution of recent years. This is because EMS are simply microcomputer systems generally used for controlling and monitoring services plants. In the case of an office building these services could be hot water, heating, lights, air conditioning and ventilation. EMS have also benefited from the knowledge and technology in the application of computer control to manufacturing and the process industry.

The earliest ancestor of the EMS was the hard-wired centralised control system. It first appeared in the 1960s and was employed in large buildings (Levermore, 1992). The system was basically an extension of the conventional control wires to a central console, with dials, indicator lights and a chart recorder, which enabled an operator at the console to monitor distant plant and to see, for instance, temperatures displayed. No computers or microelectronics were involved, and it relied on the operator to change control settings and times.

These hard-wired systems were then improved with the telephone technology of the day to enable individual items of plant to be switched. This switching occurred via data gathering panels local to the plant, into a central, multicore trunk cable running around the building from the central console. This switching, or multiplexing, saved on cabling by utilising the same trunk cable for a number of data gathering panels. Another development was the addition of a back-projected slide screen, linked in to the switching, to show the equipment status on a diagram of the selected plant.

With the rapid advances of microelectronics, and hundreds of transistor devices being integrated on to one large-scale integrated (LSI) chip, the first computer-based monitoring and control systems emerged. These early EMS were centralised energy management systems and first appeared in the 1970s (Levermore, 1992), having been developed in USA. The central station was based on a minicomputer, which contained only computing power or 'intelligence' in the system, with 'dumb' or unintelligent outstations which were boxes or cabinets for relays and connections to sensors and actuators, similar to the earlier data gathering panels (*Figure 7.4*). The central station (the minicomputer) was programmed to calculate and make decisions based on the data it received from the outstations.

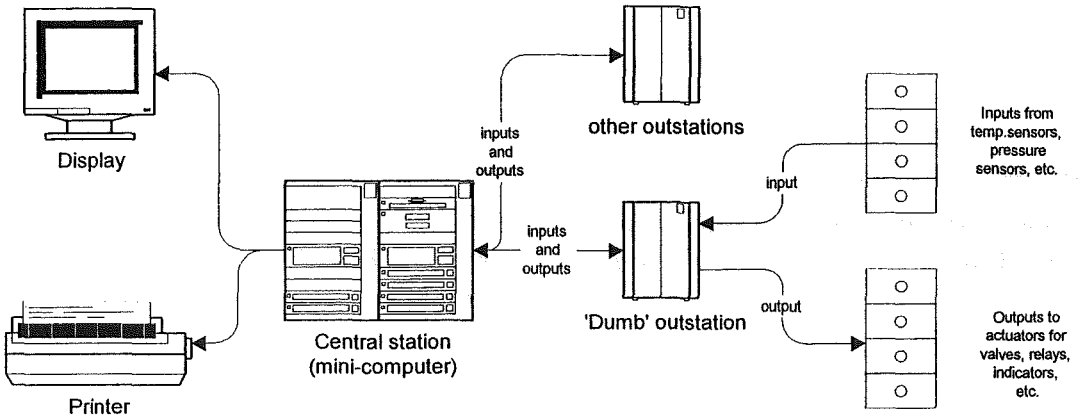


Figure 7.4 – **An early centralised Building Energy Management System.**

The systems were expensive and so viable only for large sites. Such sites were often prestige air-conditioned headquarter offices of over 12,000 m^2, or hospitals with over 500 beds, large building complexes and industrial sites and factories with over 2000 employees. Although these systems related initially to the control and monitoring of heating, ventilating and air conditioning (HVAC) plant and were therefore energy management systems, they were also capable of controlling the lighting, the lifts, and the monitoring of the security and fire alarms, although the latter two were rarely linked to building EMS in countries like the United Kingdom and New Zealand due to regulatory bodies. In fact the systems were considered as building management systems to help in the management of large and complex sites, without specifically saving energy. These early building energy management systems (BEMS) were in fact in existence before the Energy Crisis of 1973-4.

Although these early BEMS were capable of monitoring and controlling fire and security systems, they rarely did so, as dedicated systems were used for the potentially life-saving fire detection devices and also security devices. There are still problems in integrating all systems such as fire alarm systems and security systems into BEMS today, mostly due to the different regulatory bodies and standards - and to the different manufacturing companies involved - rather than the technology.

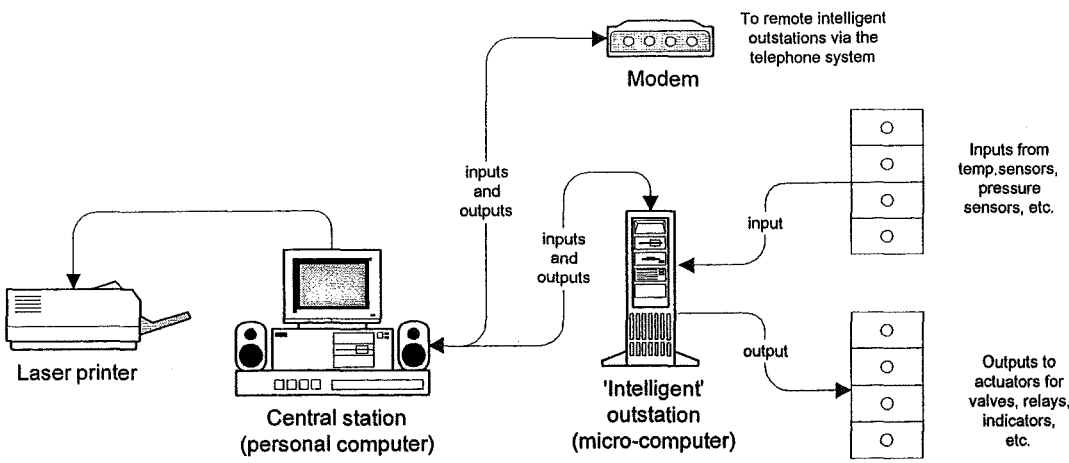


Figure 7.5 – **A micro-processor based BEMS with intelligent outstations.**

Since about 1980, with the rapid onset of LSI and very large-scale integration (VLSI) to thousands of devices per chip, microcomputers, better known as personal computers (PCs), became as powerful as the previous mini-computer, if not more so. Also the outstations,

which are themselves small microcomputers, or more correctly, they have microprocessor chips, have gained considerably in processing power (*Figure 7.5*). This enables them to operate on their own, or to become 'stand-alone' outstations, being dependent only on the central station for a small proportion of their operating time. These outstations have considerably more control functions than the older, 'dumb' outstations, which tended to have more of a monitoring role. Indeed each intelligent outstation can control a small building on its own, and it is economic to install these types of outstations in medium- and small-sized buildings. Also as a consequence of the computing revolution, the central station is based on standard PCs with sufficient memory and software capabilities. The central station can communicate with many outstations when it needs to, either on a local communications network or to remote outstations over telephone lines, as is shown in *Figure 7.5*, using modems and autodiallers.

Communications networks have now been developed to allow small buildings services, even down to light switches, with silicon chips added, to communicate.

As the power of the personal computer increases and the electronic chips become even more powerful, so BEMS central stations will become more sophisticated and small outstations for individual items of plant will become cheaper. It is rapidly becoming economic for small buildings to have microprocessor communicating devices and controls. Even sensors are becoming intelligent, or 'smart', with the sensor and chips on one piece of silicon. This intelligence, accompanied by cheaper components, has spread to smaller and smaller buildings and has now reached the domestic home. Intelligent light and power switches have been developed and standard bus systems are available.

7.5.2 EMS networks and buses

Modems can be used for communications over the telephone system for central stations and outstations. However, communications within a building can be made by a simple twisted-pair cable (often a number of twisted wires bundled together in one cable like telephone wire), connecting up the BEMS stations. This forms a local area network (LAN), or continuous bus, between the outstations and the central station. Each station needs a network adapter card (or communication node controller), to link it into the LAN. In most older style BEMS the central station acts as the master, controlling the communication. In this older, hierarchical arrangement the individual outstations do not communicate with each other, but only with the central station. A number of manufacturers do supply systems in which each outstation can act as a central station and also communicate with each other, but these outstations are much larger and consequently more expensive.

Generally, a LAN refers to the linking of a number of PCs together in a network to share high-grade, expensive equipment, such as a laser printer or a large hard disk and its large programs, so that all the PCs can use them without the need for each one to have its own dedicated and expensive equipment. Also, the PCs can share data. The PCs are linked to a LAN server, which is a more powerful PC. This server can store a number of programs that each PC can access without each needing to have a large memory like the server. So the LAN enhances the individual PCs by allowing access to expensive equipment on the network.

As BEMS outstations are effectively small computers, then their network can be referred to as a LAN, but the data transfer rate is much lower and can be slower than that required for PCs with large programs and data arrays. But similarly for a BEMS with a LAN, outstations can have the minimum memory and intelligence required, but have access to the central station with more memory and intelligence, as well as access to the other outstations. This is

very useful for large buildings, especially air-conditioned office blocks, with a number of different zones that often have separate heating and air conditioning systems.

First, consider an industrial site with a heated building used partly as offices, partly for manufacture and partly as a warehouse. One outstation could control each area with its own separate requirements. But only one outstation need have an outside air temperature sensor, solar sensor and wind sensor, as all the other outstations can access these sensors. The central station could be situated in the energy manager's office, separate from any of the outstations, but linked to them on the LAN. There could also be printer, connected to the LAN, in the gatehouse for printing alarms for a security man.

Second, consider a large multi-tenanted office block of many floors. Small outstations, each with, say, four inputs and four outputs, will control the individual items of plant (often air conditioning fan coil units or variable air volume terminal units) on each floor. Each floor would have its own separate LAN, or sub-LAN, for its small outstations. A larger outstation would be master of, or local server for, the sub-LAN and would be joined into the main LAN of the building in which would be the main central station and the other sub-LAN masters, as is shown in *Figure 7.6*.

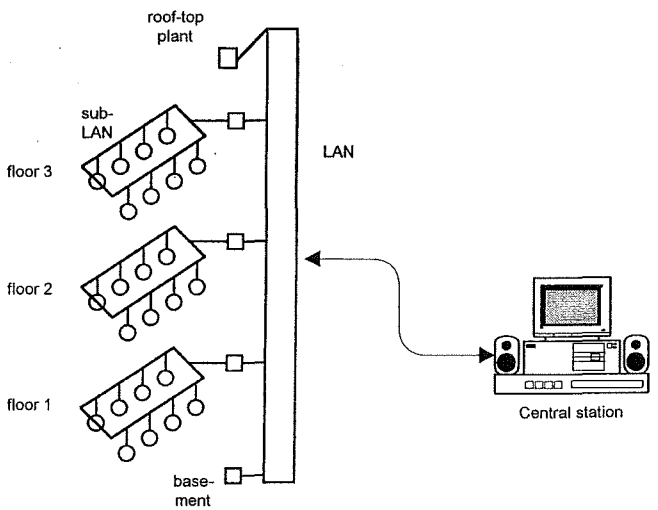


***Figure 7.6* – A large building with sub-LANs.**

Each floor's sub-LAN would control that floor's heating and air conditioning system and could monitor its energy consumption for that particular tenant. In identifying each outstation, outstations could be given attributes, such as the floor that is situated on and those that are in south-, north-, east- and west-facing zones controlling air conditioning equipment. This would allow for better control, for instance, reducing the heating in synchronisation with the sun's movements around the building when there is significant solar gain.

Small outstations are now being produced which control individual items of plant, so that the plant itself becomes intelligent. The equipment manufacturers can then put the outstation on their plant at their factory and test and commission them before they are sent to the site. Then at the outstation need simply to be connected to the LAN. Even temperature sensors, switches and relays will become intelligent (smart sensors, switches, etc.), with their own microprocessor and communications link on to a bus around the building.

There is the possibility of fire detection systems and security systems also linking in to BEMS, although failures on line system must not affect any other, making the premises unsafe in an emergency.

With this potential for communication between systems, and with outstations becoming smaller and cheaper, so that individual items of plant can become intelligent and communicate on the BEMS LAN, 'intelligent' facilities are becoming a reality.

7.5.3 Present state of the EMS and the consumer

The deregulation of the Energy Market in New Zealand appears not to have had the impact on domestic energy cost (in the form of lower electricity prices), as promised by the government when the market-change was first launched. No doubt the larger industrial consumers, or even individuals such as dairy farmers who can form themselves into some form of organised bulk buyer of energy, will find some benefit in the form of negotiated lower energy prices, but the small household consumer will find very little financial change from the pre-reformation electricity bill. It will, however, change the way the consumers view energy usage. The power of deregulation may not make as much of a difference as people think it does, but people's awareness and the amount of power they're using is going to become an important issue.

Consumers will be interested in being able to manage their electricity facilities and hence their consumption more efficiently, and they will need ease-of-use solutions to help them do that. Typically end-users, be they domestic or industrial, will want to maximise their existing facilities without having to make radical changes to their systems or substantial financial input. The point is to be able to leverage the investments that have already been made and control existing systems in the house or buildings so that the consumer does not have to make a large investment in order to achieve energy efficiency.

Large-scale consumers will need to understand when and where they are using energy. They will need to know the details of their current usage before they can negotiate deals to save on costs in a deregulated environment. The small domestic consumer does not want to be bothered by trivialities like these, and it seems reasonable to assume that the greater proportion do not have the knowledge or the means to monitor the detail-pattern of their electricity consumption.

Existing energy management systems, fitted with commercially available data gathering equipment and specialised software, should be able to furnish part of the information that the larger consumers will need. The smaller domestic consumer will find a solution in fully automated energy management systems, which act in a pro-active manner to reduce energy costs, such as the proposed FEMS.

The integration of controlling energy usage and building systems, often referred to in the literature as *Building Energy Management Systems* (BEMS), will allow such benefits as long-term cost savings and better security. A key-component to improved integration will be system *communication*; the routers for connectivity will give the ability to tie existing systems together to achieve results. An example of what would be possible with system integration for a domestic end-user, is that you could drive up to the garage of your house and insert your card into a reader. A closed-circuit television camera (CCTV) scans your face, compares it to the card, and opens the garage/house door. As you park your car, the lights in your lounge, the coffee pot, and your computer turn on, and the air-conditioning/heating adjusts to your pre-set levels. When you leave, a motion detector in your hallway or garage sends a signal to shut off your computer and coffee pot. Lights dim and the temperature adjusts to a non-occupied status. The security system knows you're gone, and it is on alert for intruders.

Such connectivity is possible today; the technology is available, the products are there to allow integration of different subsystems together that in the past have been standalone. As it stands, however, no overall *communication protocol* or standardised system has won global approval and acceptance. Two of the main protocols under consideration, especially in the USA, are *LonMark* and *BACnet*. LonMark is proprietary system that commits the end-user to one vendor: the Echelon Corporation in Palo Alto, California, which produces LonWorks. BACnet is not manufactured by one particular company, and its language has undergone a public review process. As can be expected, there are pros and cons to each protocol and both come with 500 page manuals explaining the ins and outs. One of the major differences between them is that BACnet consists of pure software and should therefore be able to facilitate the inevitable upgrades without too much bother. LonWorks, on the other hand, uses a specialised version of 'C' ("neural C") and is centred around, and is in actual fact embedded in, a so-called *neuron* chip. This could conceivably make an upgrade more expensive if not more problematic. But the neuron chip and associated transceivers offer readily available hardware and appear to make LonWorks consistent. The neuron chip, it should be stressed, has nothing to do with neural networks as propounded in the previous chapters.

The debate continues. From the point of view of the end-users, be they domestic, commercial or industrial, what is needed is a consistent communication solution, but not at the expense having to buy a proprietary system and then be committed to one vendor. After all, that one vendor might raise prices or give less-than-stellar performance, knowing the consumer is trapped. In addition, consumers would not want to pay any more than necessary to expand functionality in the future.

As just such a solution for the bigger players, manufacturers are promoting interoperability over the Internet. This raises understandable security issues given the net's open nature; a lot of companies are uncomfortable with monitoring over the Internet, but this can be weighed up against the fact that large companies, with many different offices and plants, need an efficient, effective way of communicating with their various sites. The Internet offers a cheap solution and has a rapidly developing graphical user interface (GUI) that is readily programmed to suit with commercially available web page design software. This would allow Facilities Managers across the world to quickly and easily monitor anything that's going on in any building. Additionally, using the Web, it is possible to disseminate information to anybody that may be interested. It also allows companies to modify the look and feel of the information - give managerial reports to people in the accounting office, process status reports and alarm information to engineers - all published by the same source.

7.5.4 EMS software

The newly deregulated market in the United States has seen an increase in the available energy management software packages, but again these are expensive and aimed at the large consumers. The basic information monitored by energy analysis software includes use and demand in 15- to 30-minute intervals and cost; however, more detailed information like power factor, taxes and special fees may also be tracked. A number of the software packages allow facility professionals to optimise their operations by performing proactive maintenance, accessing alarm information, obtaining real-time information remotely and dispatching engineers from anywhere. Some software versions can even compare the effects of changing rate tariffs.

Becoming increasingly popular are web-based applications, such as secure Internet sites, where facility professionals can access daily usage data — in real time — for individual or

multiple facilities and interface to existing information sources, such as meter data collection systems, energy management systems, communications systems and building control systems through third-party vendors.

This energy information and analysis software can help facility professionals monitor and manage consumption and costs through a better understanding of how their buildings and plants operate and the relationship of costs to load profiles. The software will generally provide facility professionals with reports that enable them to verify system performance. The information contained in these reports can alert them of changes that may result in higher operating costs, discomfort and breakdowns. Viable energy information and analysis software packages will play a vital role in helping facility professionals plan and manage for change rather than simply react to it. With some systems, tracking begins at the consumption point and ends at the billing system. Some of these systems take information to another level by adding real-time decision-making and accounting features. With the advent of real-time pricing, these systems could become important for the management of consumption. Some automated systems monitor and detect every opportunity to save energy — every minute, every hour, every day. Still other systems incorporate comprehensive energy management into distribution monitoring by including information obtained from meters, circuit breakers, protective relays and motor starters. Energy information software tools help facility professionals compare apples to apples and maintain an accurate record of facility costs, as well as determine how to cut costs in the future by reducing overall energy needs through efficiency measures. Facility professionals know that electricity is one of the largest controllable costs for most facilities. It makes sense for them to look at these costs even more closely under deregulation.

To assist facility professionals with data tracking, some electricity retail companies will provide energy bills with historic information in electronic format, so the information can be automatically imported into the tracking database, rather than entered manually. Even though various types of energy information and analysis software have been around for 20 years, they have only become popular with the advent of deregulation. Currently, less than 5 percent of all commercial buildings in a country like America employ energy information and analysis software products.

Analysts predict that as the cost of hardware for on-site control continues to drop, and industry standards for control increase, the market for energy information products will grow. This is good news for both the consumers and companies that deliver systems akin to FEMS.

Chapter 8. FEMS: a Fluid Energy Management System

8.1 Reviewing the basic principles

Many existing hot water cylinders are set around 65 to 75^{0}C. From a safety aspect this is considered too hot for water to be used directly. Reducing the temperature to a safer level will not only prevent many accidents but will make the hot water cylinder more efficient because less energy is required to heat the water to its maximum temperature. The standing heat loss is reduced since the water is stored at a temperature close to that at which it is to be used. Depending on the insulation, lowering the water temperature from 65^{0}C to 55^{0}C will reduce the cylinder standing heat loss by 20%, but will also reduce the effective capacity by 40%. The capacity of a system can be taken as the amount of water it can provide above 40^{0}C (typical shower/bath temperature) after mixing the hot water with ambient cold water.

Energy efficiency in providing hot water services extends beyond the issue of just heating and storing water. It also depends on consumption levels and patterns. Hot water demand by individuals can vary significantly, even on an hourly or day-to-day basis. Maximum system demand (litres per hour) is typically twice the average demand (CAE, 1996). Factors such as increased numbers in the home, successive loads of washing and varying incoming water temperatures (summer/winter) all impact on total demand. Houses change hands and family sizes vary - all of which can alter a given usage pattern. The most common way to accommodate these factors is to adjust the cylinder temperature setting by means of the thermostat. It will not always be necessary to heat up 180 litres of water to, say, 75^{0}C. If only 30 litres of the water is used in the course of a day for washing hands or cleaning dishes then the remainder of the stored heat will slowly dissipate to the environment and to the bottom layers of colder water (the rate of heat loss to the environment is largely governed by the type and thickness of the insulation).

If prior information was available on the quantity of hot water needed to meet the demands of the next 24 hours, then dedicated equipment could ensure that only a volume of water (plus a reserve) is heated that is estimated to be required. This would constitute a significant saving in the charges on an electricity bill.

This points the way towards an *intelligent* energy management system which is capable of learning the day to day hot water requirements of any given household, and which can predict future hot water demand with a degree of adaptability. The Fluid Energy Management System (FEMS) as described in this chapter operates on this principle.

8.2 Local distribution authorities and night-rate tariffs

In addition to modifying the heated volume of water, which represents both an energy and cost saving, an additional financial saving can be made in terms utilising cheaper electricity rates. A typical hot water draw-off pattern is shown in *Figure 8.1*, note the peaks around 7 in the morning and 9 in the evening (Hendtlass, 1981). It also clearly shows that the consumption of water and the associated electric heating occur on an intermittent basis. As the monthly electric bill can typically have 35 to 45% of its charges attributable to the HW cylinder heating it makes good financial sense to reduce the bill by heating the water at a cheaper tariff rate if one is available.

At present in New Zealand such a tariff rate is available for domestic users during the off-peak night hours, typically 11 p.m. to 7 a.m. This off-peak tariff can cut the kWh charge by up to 50% and is the main contributor to the economic heating of stored hot water. There is a cost benefit for the electricity supply authority as well, in that it reduces the peak load during the day-time, and helps towards spreading the total load more evenly in a 24 hour period.

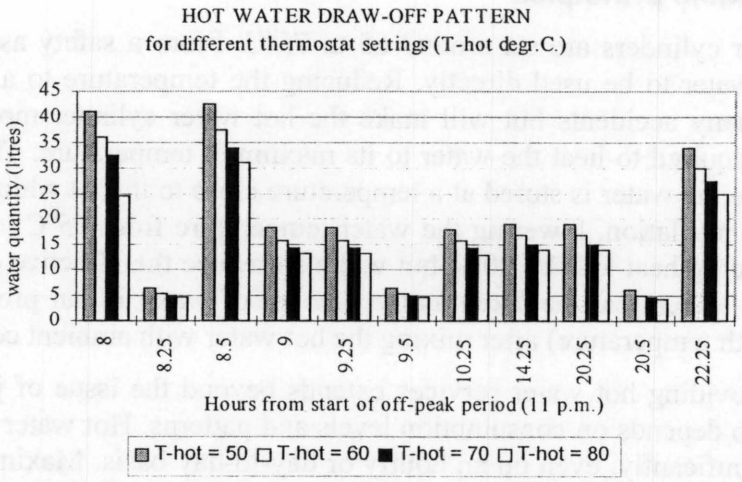


***Figure 8.1* -Typical water demand profile.**

If water is heated using the off-peak night rate then the supply authorities ensure that no heating takes place during the hours other than 11pm to 7am. This control is achieved by activating a ripple relay, usually installed in the household switchboard. The relay switches the heating element on or off and is activated by a 'ripple' signal (the local supplier uses 175 Hz), which is superimposed on the mains supply frequency of 50 Hz by the supply authorities.

The result of this limited heating period is that the HW cylinder will be required to store enough hot water to meet the total daily demand, as there is no way of replenishing the hot water if the total volume of 180 litres is drawn off and more is needed. The cylinder volume being a fixed parameter means that the consumer can only vary the temperature setting if he wants to achieve some form of control. In practice however even this setting is hardly ever altered.

8.3 System overview

The scope of the Fluid Energy Management System presented in this chapter incorporates both prediction of the next day's demand and utilisation of the night rate tariff. This system, FEMS for short, consists of both hardware and software.

No direct input from the domestic consumer is required. The FEMS system intends to be as intelligent and fully automated as possible; the only possible feedback of interest to the consumer might be the diagnostic and consumption messages when requested. Electricity used by the cylinder is monitored and the total consumption can be displayed, along with costs, with the push of a button.

8.3.1 software

The software part of the system is build up around a real-time kernel. The kernel delegates the tasks such as the monitoring of the temperature sensors and storage of data at the correct time. The program manipulates the data to build up a time series of the necessary and possibly relevant data such as historic hot water demand, draw-off patterns, time of day, day of the week, month, periodic and non-periodic holidays, and outside temperature. Not included but of relevance in more tropical climates would be the humidity. These various time series are used to train *two* artificial neural networks (ANN); *learning* takes place in an *off-line* manner. This means that at a set time once every 24 hours the latest historical data is presented to the relevant ANN inputs for learning purposes. For the FEMS this set time is just after 11pm, the onset of the cheap night-rate period. The aim of this exercise is to build up a model in the dual neural network of (i) the total hot water energy consumption and (ii) the times of the first major draw-off. The reason for wanting to know the latter information is explained below.

The artificial neural network catering solely for hot water demand has a single output. The output constitutes the hot water demand, or more correctly the energy demand, predicted for the following 24-hour period. With this value known the system calculates the length of time the cylinder's heating element needs to be switched on in order to be able to add the predicted energy requirement - via the element - to the water stored in the cylinder. The system will attempt to have this energy addition completed at a time *as close as possible* to the time of *first* major hot water use, but still within the period allotted (i.e. 11pm till 7pm) for cheaper night-tariff rates. Usually the most common cause of first usage is an early morning shower taken by a person preparing for work. In this way the management system minimises heat loss to the surroundings, which inevitably occurs in almost every domestic hot water system in New Zealand where the water is heated either continuously or, for those on night rate, where heating commences at 11pm each day.

The software therefore needs to monitor and store not only the daily hot water energy demand but also the *time* of *first* major hot water draw-off. The second ANN thus needs to make a prediction of the first draw-off time.

Figure 8.2 shows a flow-chart of the major software activities. Processor intensive are the ANN algorithms. ANN learning takes place at midnight, as does the forecasting of both hot water demand and time of draw-off. The software uses ANSI C in order to ease the transition at a later stage to a dedicated microprocessor. For the purposes of the test-trials a Dos and a Windows interface (GUI) program were specially written.

A detailed description of the neural networks utilised for prediction is given in *Section* 8.3.3.

The structure and contents of the FEMS total software package is explained in greater detail in *Section* 8.4.

8.3.2 hardware

For the purposes of gathering data on *actual* household hot water demand (as per what a commercial version of FEMS would collect), the hardware used consisted of the tried and tested components first encountered in *Chapter* 2. This included all the equipment then used, bar the 240VAC auxiliary relays (as there was no need, at this stage, to control the power supply) and had as an additional component a thermistor for measuring ambient (attic) temperature. The complete installation can be listed as follows:

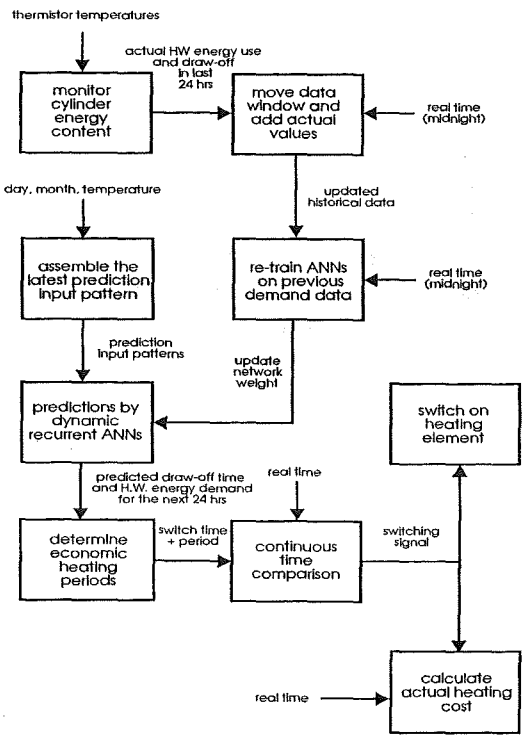


Figure 8.2 **– The basic software activity flow-chart.**

The hot water cylinder – 180 litres, medium pressure, installed in the attic of a typical domestic household consisting of two adults and two children aged 7 and 10. For the purposes of data gathering the upper element was disabled.

The thermistor strip - it consists of 19 negative temperature coefficient (NTC) thermistors whose resistances drop in a non-linear exponential manner as temperature increases. They are connected in series and the voltage drop across each thermistor forms the input signal for a dedicated A/D converter port *Figure 8.4.*

The interface/supply board- this board contains the constant current source, a binary counter (for the flow-meter) and a number of small relays. It acts as central junction box for the various connections to and from the rest of the equipment.

The flow-meter – not strictly necessary but used to confirm and sometimes clarify the readings obtained from the temperature sensors installed on the cylinder.

The constant current supply - a 0.2 mA current from a constant current source flows through the 19 thermistors *plus* one other thermistor used to measure the ambient environment temperature.

Ambient temperature sensor - located on the interface board it provides data thought necessary to improve energy usage prediction.

IBM compatible PC – a 486/33 based computer system, which includes the data acquisition cards as described in detail in *Chapter* 2. The A/D section features 2.5 VDC differential input ports with 12 bit resolution.

It is the intention that a *commercial* version of the FEMS should be able to be readily installed on any existing domestic hot water cylinder. The system would probably consist of five main hardware sections; a cylinder temperature monitoring device, an interface/supply board, a 240V AC relay, and a processor/memory board. The function of each is as follows:

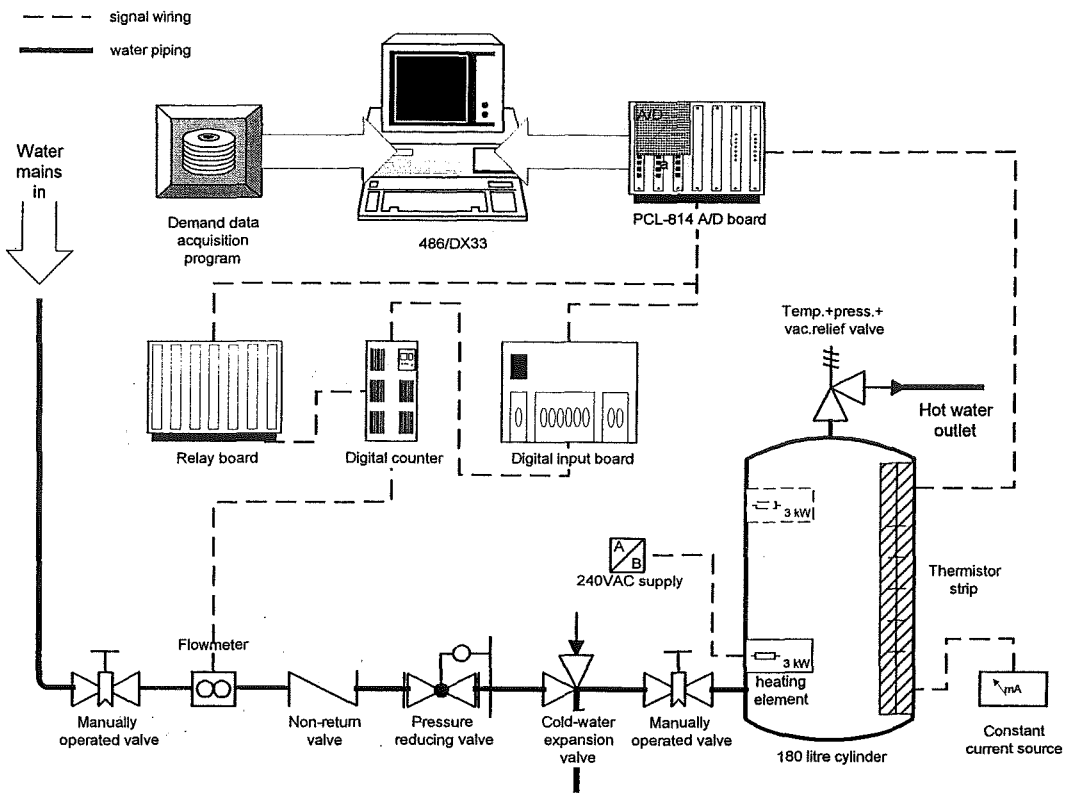


***Figure 8.3* - Overview of the FEMS equipment as used for demand data collecting.**

Temperature monitoring device – this is equipment akin to the thermistor strip used in tests of *Chapter* 2. Despite its cost-effectiveness, the presence of thermal lag, the analogue signal output and the inherent non-linearity does not make the thermistor an ideal tool; some of the presently available ICs specifically designed as temperature sensors are deemed more suitable.

Interface/supply board – The selected monitoring sensors in whatever form will need a power supply, likely to be +5V in case of ICs, and a means of inputting the sensed temperatures to the microprocessor board.

240V AC relay – The system must have a mains voltage relay as a means of controlling the power supply to the heating element. The relay would be operated by a micro-processor's DO (digital out) signal.

Micro-processor/DSP and memory circuit board – This dedicated processor/DSP with its associated memory banks and small display panel would replace the PC as used in the test systems. Greatly dependent on the complexity of the final commercial design in terms of the number of neural units, input and output signals, display capabilities, and the size and number of software modules, this board forms the most expensive part of a commercial version of FEMS.

What is touted as being of major advantage in commercially available devices is also applicable here: the FEMS is non-intrusive so there should be no interruption to the hot water supply during system installation.

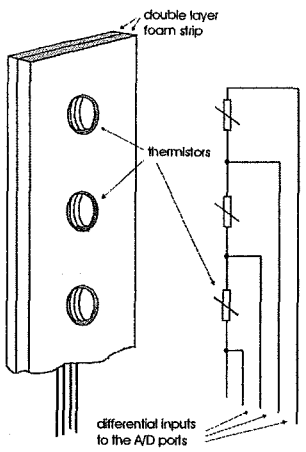


Figure 8.4 - **The 19 NTC thermistors are pasted on a flexible foam strip and are pressed against the copper skin of the cylinder.**

8.3.3 Prediction - the neural network

The crux of the system is being able to reliably forecast the hot water use at the place of installation. When the system is powered up for the first time no historical time series data on water usage or draw-off time is available. A minimum amount of data will therefore need to be collected before operation proper can start. As an alternative a typical standard set of data could be used as a starting point.

The intention of utilising a small quantity of historical time series data makes it difficult to use classical forecasting techniques such as Box-Jenkins, k-nearest neighbour or Kalman structural methods as these require a considerable amount of explicit historic temporal sequence data to build effective models. Other disadvantages of these methods are that they do not readily allow easy modelling of a non-linear multi-variate time series; stochastic methods would not handle a sudden water consumption increase because they consider it as noise, statistical methods are based on feature space obtained by linear analysis, and the k-nearest neighbour rule needs to keep in memory a large number of examples; it has no memory loss and therefore does not evolve (Canu, 1990).

A considerable amount of recent research has shown that artificial neural networks (ANN) perform as well if not better the same tasks normally handled by the classical methods. They can be of an adaptive design and are easier to set up and implement than their classical counterparts. Other important advantages for neural network prediction learning are that little pre-processing is necessary, non-linearity is easily modelled, and training examples can be taken directly from the temporal sequence of the input; no special supervisor or teacher is required. There is also no need to store all past data as the information is contained in the synaptic weights of the network. However, the lack of historic data forces a judicious choice on the type of network to be employed and the training methodology needed (Wezenberg et al, 1995).

After considering the merits of a variety of neural networks the final choice settled on a 3-layer Elman recurrent network with a number of feedback loops (Elman, 1990), (see *Figure 8.5*). The Elman Recurrent Neural Network offers improved spatio-temporal modelling and Elman has shown that recurrent connections allow the networks hidden units to see its own previous output, so that the subsequent behaviour can be shaped by previous responses. The recurrent connections act as memory and allow time to be represented implicitly by its effect on processing rather than explicitly (as in a purely spatial representation).

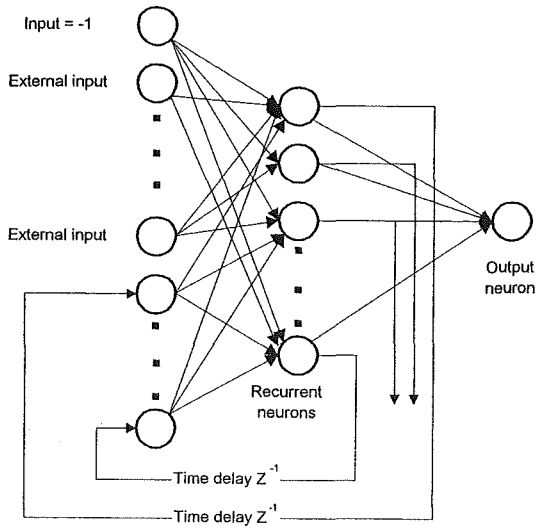


Figure 8.5 - **An Elman recurrent neural network.**

The recurrent neural networks used in the FEMS consist of an input layer, a recurrent (hidden) layer, and a single output neuron. The neurons in the recurrent layer have a transfer function with sigmoid characteristics. The numeric output from the hidden layer is fed back, with a *single* time delay; note that the feedback connections carry a unity weight and are not altered during the learning (training) phase, and as such do not influence the numeric values.

The output layer has a neuron with a linear transfer function. This combination of sigmoid and linear allows the network to approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons (Demuth et al., 1995).

The input layer is split up in to a number of sections. One section consists of up to 14 nodes that input the scaled historical values of either energy or draw-off time (depending on what is being predicted), as derived on a daily basis over a two-week period. The historical values are typically scaled to between 0 and 1 so as to fall within the working range of the neuron transfer function.

A second section has 6 nodes and the scaled input values also range from 0 to 1, as per the previous section. The values describe the daily maximum and minimum ambient temperatures over a 3-day period.

The third section contains 8 Boolean nodes which input the binary representations of the day of the week, the bi-monthly period of the year (e.g. 1 0 0 = feb/mar, 0 1 0 = apr/may, etc.), and the type of holiday. The first of the last two Boolean nodes is set to 1 if the day of prediction is a regular holiday, such as Christmas, and the last node is set to 1 if we're expecting an irregular holiday. Days such as ANZAC and Easter are considered irregular holidays as they are not fixed to a specific date (Walkington, 1989).

In the learning phase the neural net is repeatedly presented with a set of the latest stored historical data (the time series) in the form of *vectors*. Each vector contains a single set of input data equivalent to the nodes in the input layer of the neural network.

The training set starts with a minimum of 2 weeks worth of daily *in-situ* collected data. These two weeks worth of the historical time series forms the very first training vector with 14 contiguous days of energy or draw-off time values. The actual learning process itself starts two weeks later after that first stage has been reached, when a minimum 14 vectors is

available in the training set. Thus at least 4 weeks need to pass before the first prediction can be made. The training set builds up to a pre-set (see *Section* 8.5) maximum number of weeks of stored data as time progresses and further pattern information is collected. When the maximum figure is reached and exceeded the system software sets up a data window, which moves on a daily basis. With only the latest data being used and re-training the ANN every 24 hours the effect is that the system can incorporate any new changes in its network parameters; in effect it adapts itself to a new situation or environment.

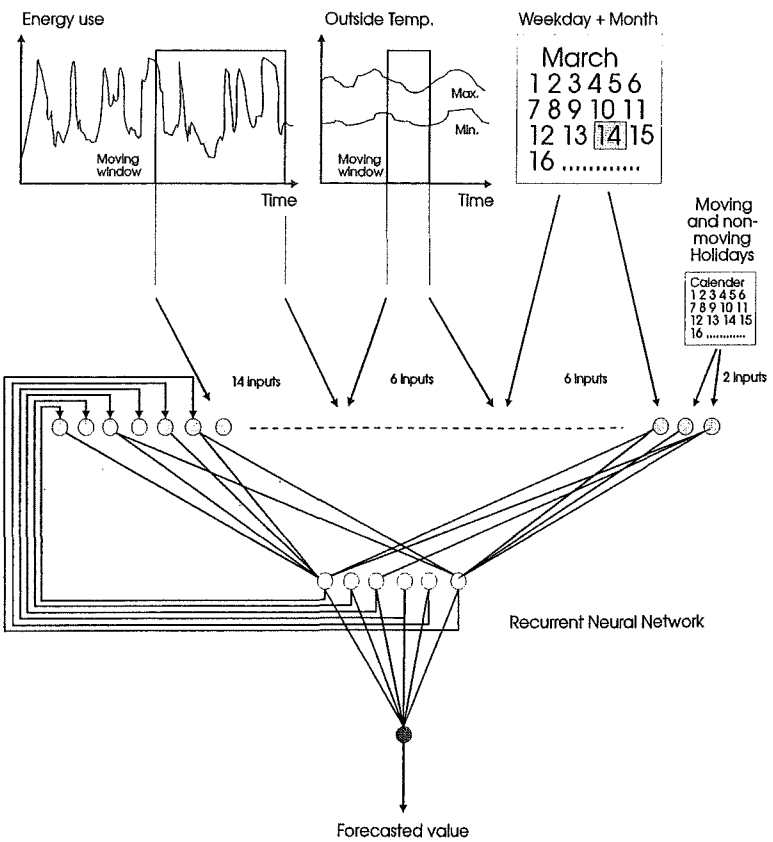


***Figure 8.6* - The Recurrent Neural Network and the data inputs.**

8.3.4 The historic data input vector and the prediction data input vector

The data used for training the neural network prior to letting it make a prediction for the next day, a process repeated once every 24 hours, differs in both the quantity of data presented as input and the make-up of the input vector.

Figure 8.7 displays the properties of a *single* historic data input vector, which, from this point forward, will be referred to as simply a *training vector*. As there are two different neural networks there will be an *energy training vector* and a *draw-off time training vector*. *Section* 8.3.3 has already mentioned the three different sections of the input nodes for the 'energy prediction' and 'first draw-off time prediction' neural networks. The energy training vector must reflect these same sections, and as such the values of the daily total used energy for the past two weeks (counting back from the day when $E_{total(0)}$ was derived) make up the first part of the vector and are labelled $E_{total\ (14)}$ to $E_{total(1)}$.

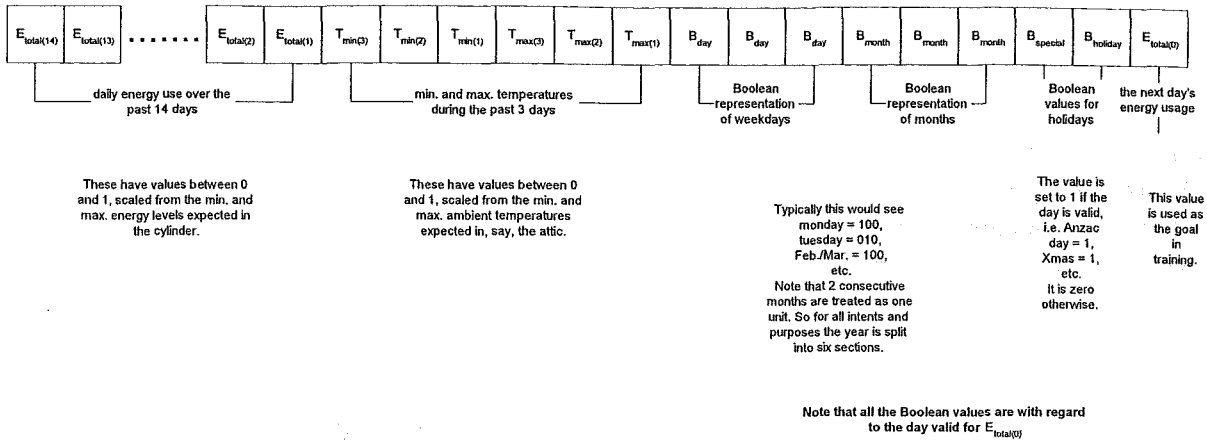


Figure 8.7 **– The energy training vector; being an example of the sequence of data presented to the input of the neural network(s) for off-line training**

A similar situation exists for the draw-off time prediction vector (*Figure 8.8*) where the variables are labelled $D_{time\ (14)}$ to $D_{time\ (1)}$. The number in the subscript of each variable signifies in days how old it is in relation to $E_{total(0)}$ or $D_{time(0)}$; $D_{time\ (1)}$ is therefore the previous day's draw-off time and $D_{time\ (14)}$ occurred two weeks earlier. The figures in these variables are not the actual values for energy and time but rather have been scaled between 0 and 1 to an accuracy of 4 decimal digits after the point. The representative range for each parameter is shown in *Table 8.1*.

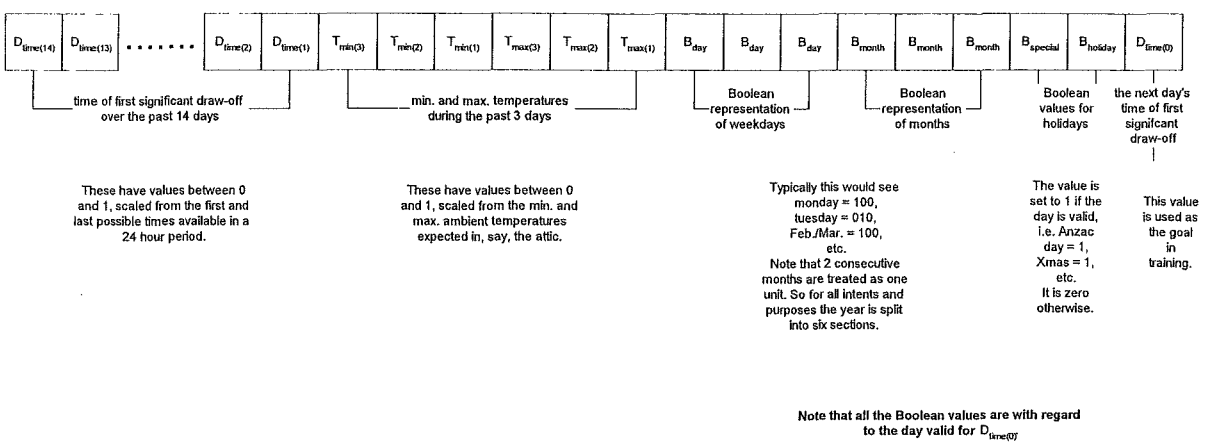


Figure 8.8 **– The draw-off time training vector; being an example of the sequence of data presented to the input of the neural network(s) for off-line training**

The next 14 variables are common to *both* the training vectors. The daily minimum and maximum temperatures over the past three days are presented in $T_{min\ (3)}$ to $T_{max\ (1)}$, giving six values in total. There are also six Boolean digits, three for the day of the week and three for the dual-months. Both the day and the month pertain to the day that $E_{total(0)}$ or $D_{time(0)}$ occurred. The Boolean equivalent for the day and dual-month is shown in *Table 8.2*. The reason for joining two consecutive months together is that it reduces the number of Boolean inputs needed c.f. a single month representation; as the seasonal difference between consecutive months is on the average small, it should not result in giving contradicting signals to the network for any of the variables used.

It has been mentioned more than once already that forecasting models, dependant on or influenced to some degree by irregular and/or seasonal variables, do well to incorporate these influences in a relevant manner. It is reasonable to assume that some of the households will alter their consumption of hot water as a direct result of it being a non-working day; and it could certainly influence the time of first significant usage. Research has shown (Walkington, 1989) that overall power consumption is definitely altered by these factors; which then becomes quite significant when variable tariff rates are included in the enhanced version of the FEMS (*Chapter* 9). It is therefore prudent to attend to this matter now, and as such the training vectors have been given two Boolean operatives. One is for the regular holidays and the other for the irregular days, such as ANZAC and Easter.

Temperature range (^{0}C)	Energy range (kJ)	Time range	equivalent scaled values
10 – 90	0 – 54000	00:00am – 12pm	0.0000 – 1.0000

Table 8.1 – **The ranges for temperature, energy, and time and their equivalent scaled input for the neural network.**

The 29th and last value in the input string is $E_{total(0)}$ or $D_{time(0)}$; this is the actual training goal and is only used for comparison, not as an input. This means that when a single sequence of the training vector is fed into the 28 node input (actually 29 nodes, as a bias value of –1 is also included) of the neural network the resultant single figure output should be as close as possible to the goal as required. Any resultant error will determine, along with all the other generated errors, whether the training is repeated or whether the neural network is deemed sufficiently trained.

Boolean code	day	months
100	Monday	Feb./Mar.
010	Tuesday	Apr./May
110	Wednesday	Jun./Jul.
001	Thursday	Aug./Sep.
101	Friday	Oct./Nov.
011	Saturday	Dec./Jan.
111	Sunday	-

Table 8.2 – **Equivalent Boolean code representing the day and the month as used for a section of the historic data input vectors.**

After the software has generated the latest training vectors and added them to the training files (see the data-flow section 8.4.2) it is time to construct the two *prediction input vectors*. These vectors are basically the same as the latest version of the training vectors, except that the most *recent* $E_{total(0)}$ or $D_{time(0)}$, being the value of the respective training goals, is moved to the historic data section of the input vector and takes the place of $E_{total(1)}$ or $D_{time(1)}$. All the other historic values move down one position, which means that the old $E_{total(14)}$ or $D_{time(14)}$ drops off. A similar operation occurs for the latest values of the minimum and maximum temperatures, and the Boolean values for day, month, special day and holiday are set to

reflect the very day of the prediction. Thus, if a Monday evening forecast were made for the energy usage on Tuesday, then 010 = Tuesday would be the Boolean value in the prediction vector. *Figure 8.9* illustrates these changes for an Energy prediction vector; the Draw-off time prediction vector is, of course, similar in appearance. Rests only to mention that the wheel has turned a full circle when at the close of the *predicted* day the prediction vectors can become the latest historical input vectors, simply by adding the *actual* values for 'total used energy' or 'first draw-off time' on the end of the vector as the 29th value.

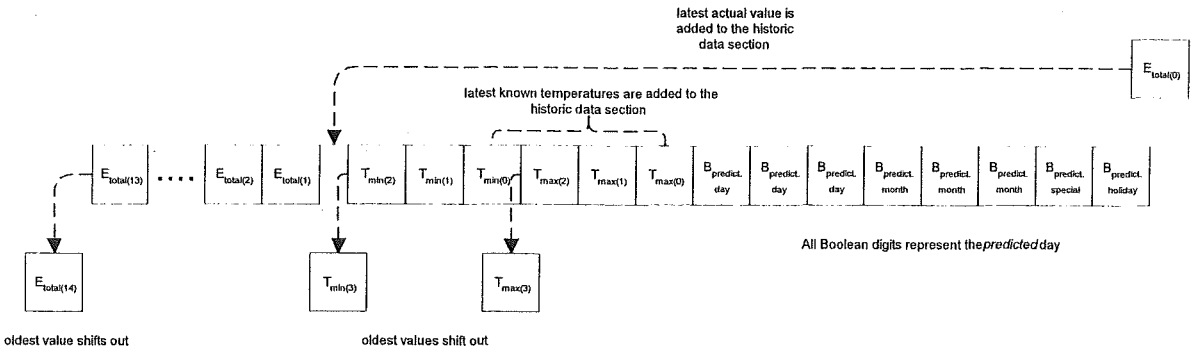


Figure 8.9 **– Illustrating the changes that take place in the most recent energy training vector to derive the energy prediction vector.**

8.4 The FEMS software in detail

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people and machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later (Brooks, 1987).

8.4.1 The design technique

The requirement analysis stage is an important and difficult stage in the development of software. There is much information that must be gathered and many communication barriers to overcome. Unfortunately, there is no foolproof way to complete this task. Several techniques or guidelines have been proposed in the past, and new ways are being developed even now. The choice of which technique to use is an important one. Some techniques were created to assist in the development of system which are useful for modelling real-time systems. Other techniques were created to assist in the development of systems where the structure and interrelationship of the data elements in the system is the central problem. Examples of these techniques include the JSD method (Jackson System of Design) and the LCP method (Logical Construction of Programs); both are discussed in Mynatt (1990). The method chosen to generate all the software for the project part of the thesis is the *Structured Analysis* method, developed by DeMarco (1978). The Structured Analysis method is most useful for more traditional data processing systems. Such systems can be characterised as ones where data flows among activities or functions in the system. The activities create or output data, or transform the data into new forms and pass along the changed data.

The structured Analysis method was selected for several reasons. First, it is one of the most widely used methods. Second, the majority of system development involves data processing systems. Third, the author has personally found that it facilitates the (inevitable) later stages

of alterations/updating. And finally, although none were used, a number of automated software engineering tools exist that have been developed based on this method (Mynatt, 1990).

In the structured analysis method, the main goal is to produce various models of the current system and the proposed system. Models are used in many different endeavours as a way of compacting and organising ideas and information. In structured analysis, the models that are built are models of systems, rather than of physical objects.

A 'system' is defined here as being a collection of interacting elements that perform some function or functions aimed at achieving some objective. In general the systems studied during software development involve software, hardware, people, physical entities, data, procedures, and documents. Because so much detail is involved in even a simple system, it is useful to come up with a model or several models of the system to help organise the details. In structured analysis the system model consists of three components: *data-flow* diagrams, a *data dictionary*, and *activity specifications* (Mynatt, 1990). Data-flow diagrams are large-scale graphs that diagram activities, data stores, and the flow of information or entities within the system. Activities and data stores are represented as labelled nodes in the diagrams. The activities are connected by lines that represent the relationships among the various activities. The lines are labelled to show what pieces of information (data) or entities are moving among the activities in the system. A data dictionary is a list of all the data elements shown in the data-flow diagram, along with a definition or description of the data elements. Activity specifications are detailed descriptions of each of the activities (nodes) indicated in the data-flow diagram.

8.4.2 Data flow

It is a fundamental concept of the structured analysis technique that a software system such as that represented by FEMS is best tackled by *decomposition*, otherwise known as *step-wise refinement*. The division into sub-problems is not arbitrary, and it is important that:

- Each sub-problem is at the same level of details as the other sub-problems it goes with (this is not as straightforward as it might first appear, especially when working several levels lower with the 'sub-sub-...-problems').
- Each sub-problem can be solved as an individual problem.
- The solutions to the sub-problems can be combined to solve the original problem.

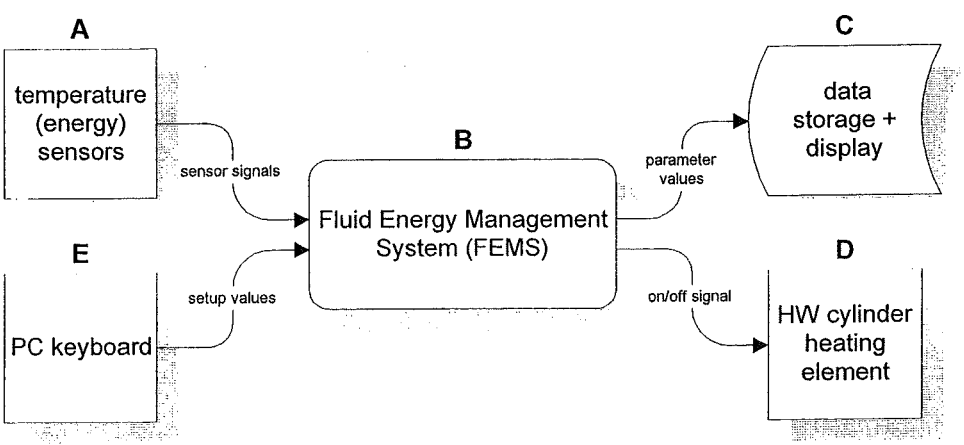


Figure 8.10 – **The fundamental data-flow diagram for FEMS.**

At the top most level the Fluid Energy Management System can be reduced to a basic input/output data-flow diagram as shown in *Figure 8.10*. The diagram also serves to introduce the convention that will be used from here on forward. *Activities* or *processes* are represented as round-cornered rectangles; an example of this is block **B**. A title or a phrase is written inside the rectangle to describe the activity being represented. Arrows are labelled with the names of the *data* entities or *materials* that are travelling along them. At the lower levels, data names that consist of multiple words will be hyphenated to indicate that a single entity is being named. A *file* or *store* of information or materials is represented using centre-skewed rectangle surrounding the name of the file or store, e.g. block **C**. The ultimate *destination* of the information or materials produced by the system (the data sink) and the originating *source* of the information or materials used by the system, but that are outside the system (data sources), are represented as squares. An example of this type is the temperature sensors of block **A.**

While building on the explanation given in *Section* 8.3.1 of the various functions assigned to the software, a fuller picture emerges when the level 1 data flow diagram is examined more closely (*Figure 8.11*).

The first thing to notice is that the concept of decomposition has been applied. As such the need to keep track of the higher-level originator activity is satisfied by labelling each sub-activity with the originator activity's tag and an additional number. Block **B** from *Figure 8.10* thus splits up into **B.1**, **B.2**, ...etc.

The basic concept of the FEMS programming is best understood by following the various data flows. Starting with the *voltage signal* from the thermistor sensors in block **A.1** the equivalent *temperature* read by each sensor is used to determine by how many kJ the *energy* content of the cylinder has in- or de-creased in the last half-hour. This means that the previous energy values need to be stored in memory in some form or other, for comparison and calculation purposes. Exactly how this storage is achieved is a step-wise refinement of activity **B.2** and represents a lower level again; it is therefore not seen in *Figure 8.11*.

At 11pm, this being the usual start of the lower electricity night rate, the total amount of energy used during the previous 24 hours is calculated (**B.3**). There is no need to reset (to nought) the 48-cell array that stores the 48 half-hourly energy figures. The new energy values that become available, with every passing 30-minute interval, simply push the oldest figures out of the array.

The newly calculated *total energy* value is passed on to the training data file (**C.2**) for later utilisation by the artificial neural network (ANN). This same file also stores the *maximum* and *minimum ambient temperatures* reached over the same 24-hour period as provided by activity **B.9**. The initial input for **B.9** has come from the temperature conversion activity **B.1**, which outputs the *ambient temperature* reading (and the cylinder temperatures) on a continuous basis. Each new reading is compared with two temperature values already present in memory: the highest and lowest ambient temperatures reached in the 24-hour period. If it exceeds either value than it replaces it in memory. Both memory values get set to zero when a 24-hour period is completed and the values have been passed on to the training data file.

The half-hourly used energy values are also monitored by activity **B.8**, but only in a time interval stretching between the hour of 12pm and the time when the decision is made that a large amount of energy has been withdrawn from the cylinder.

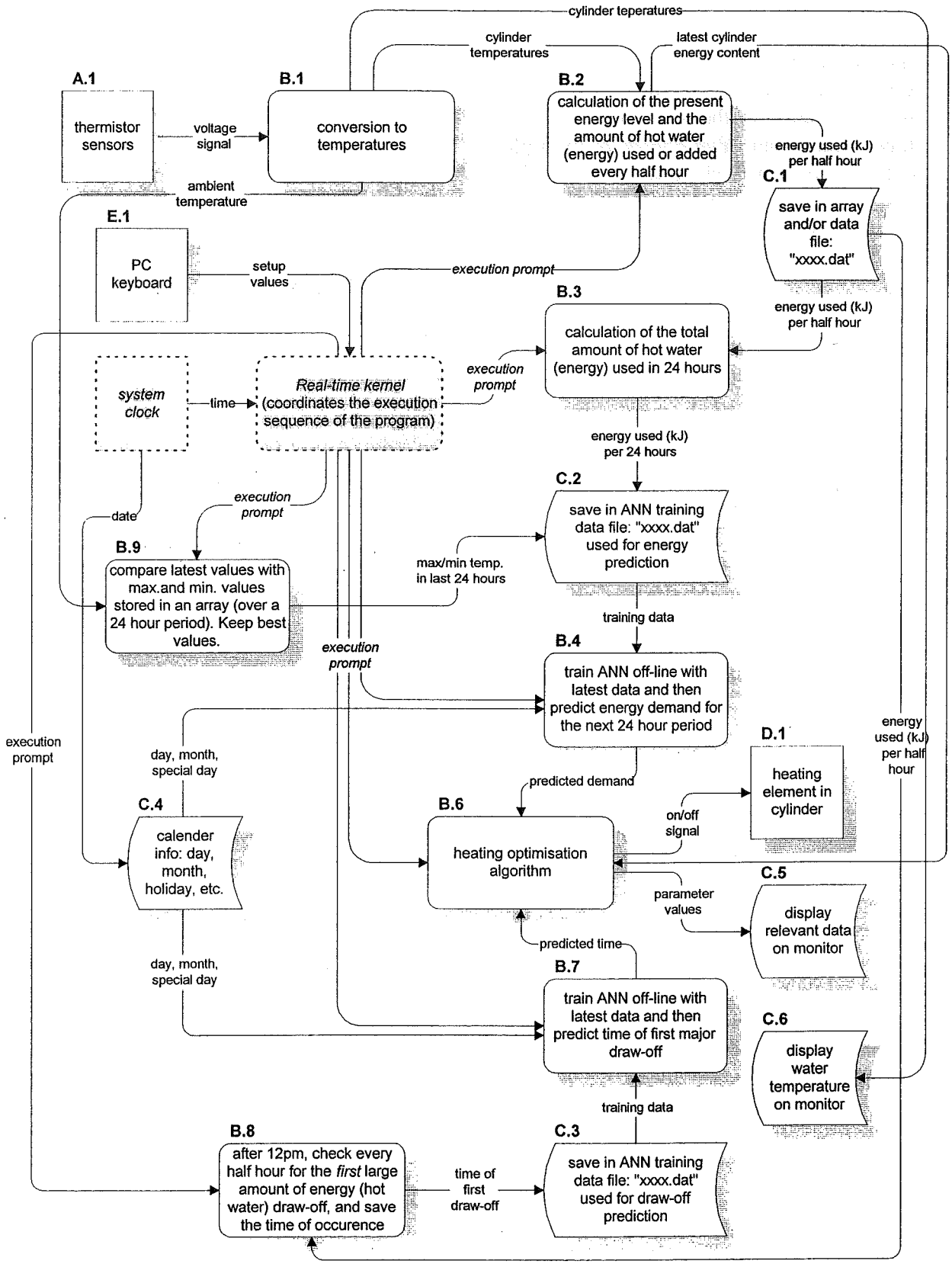


Figure 8.11 – **Level 1 data flow diagram for FEMS**

This '*time of first major draw-off*' is only accurate to the nearest half-hour, but this is reasonable given that a large withdrawal will not usually occur instantaneously but rather over a period of 5 to 10 minutes, the duration of an average morning shower. Again this data is passed on to ANN training data file (**C.3**) but this is a separate file from the training data file mentioned previously.

However, both training data file activities **C.2** and **C.3** collect input from activity **C.4**, whose function is to store the *date* and additional calendar related information. Once every 24 hours, the date, pertaining to the *day* and *month* of when the latest information is added to the data files, and a flag indicating whether the day was special in the sense of it being a regular or irregular holiday, is added to **C.2** and **C.3**.

At 11pm in the evening, when the night storage comes into effect, activities **B.4** and **B.7** are activated by the real-time kernel. It is the function of these activities to train the respective neural networks with the information stored in the data files. These files, incorporating the latest information, will train the networks until the error has been minimised to a pre-determined level (for this project the value of the level was a 'set-up' parameter, entered via the keyboard **E.1**).

The neural networks are trained prior to making the 'energy usage' and 'time of first use' predictions for the forthcoming 24 hours. These 'next day' predictions are made by presenting up to 2 week's worth of the latest data to the input of the networks (a '*data input vector*') and monitoring the resultant single valued outputs. It is important to understand that the input data and output prediction variables will all be values between 0 and 1, and as such need up- or down-scaling prior to being interpreted or put to use in some other manner by other FEMS activities. The data outflow of blocks **B.4** and **B.7**, *predicted demand* and *predicted time*, have therefore been adjusted prior to utilisation by the heating optimisation algorithm.

It is the function of the heating optimisation algorithm, block **B.6**, to make a twofold calculation: (i) how much heating time is needed to obtain the required energy level in the cylinder (i.e. the predicted energy) and (ii), when should the heating element be switched on?

The length of the heating time is determined by the amount of energy remaining in the cylinder and the value of the prediction. Thus:

$$\textbf{Energy to be added} = \textbf{Predicted energy} - \textbf{Remaining energy} \qquad (8.1)$$

If the remaining energy *exceeds* the predicted value then all the energy that is required for the forthcoming 24 hours is already available and the element need never be switched on. But if, as is most likely the case, the available energy is *not* enough to meet predicted demand than the algorithm must determine the length of time the element will need to be heating the water so as to obtain the required energy level as per equation (8.1). Thus,

$$\textbf{Total heating time} = \textbf{Energy to be added / Rate of heating} \qquad (8.2)$$

For the element in the test cylinder the rate of heating was obtained from the test trials in *Chapter* 2 and determined to be an average 168 kJ per minute; equivalent to a 0.23°C per minute cylinder water temperature increase. The value is an average, as the rate of heat loss to ambient will affect this figure to some extent. For a commercial system this set-up value will be determined by the size of the heating element and can either be resolved by the FEMS *in situ* or be input by whoever installs the system. From the viewpoint of accuracy and ease of installation the first is the preferred method.

As to when the element should start to heat the water, this is derived from knowing when the first major draw-off is likely to take place and the length of the heating time. A simple

calculation suffices to give the start and end time of the - to be applied - energy. Should the end time occur *later* than the end of the night rate period (7am) than the start time is adjusted (moved to an earlier time frame) so that the complete heating cycle falls within the cheap period.

The output from the algorithm is a simple *on/off signal* directed to a relay, or set of relays, which switch the power to the heating element as represented by **D.1**.

C.5 and **C.6** are there to provide feedback on what is happening with the system by displaying the water temperatures at various levels along the length of the cylinder as well as pertinent information on several variables used or output by the system. Although not strictly necessary in a black box system approach to FEMS it is hard to bypass the fact that even a fully automated system needs to give an indication of its present status to the owner, if not certainly to the person responsible for the installation and proper functioning of the system.

In this category would fall any diagnostic messages as well as the water temperatures. But even if this were not the case, the fact remains that a display of the relevant system parameters is invaluable at this stage of the project, when there is a clear need to know what is happening at the various operational stages and times of the day.

A typical example of a Level 2 data flow diagrams can be seen in *Figure 8.12*. Activity **B.6** was selected to demonstrate the step-wise breakdown involved in the 'heating optimisation algorithm'. Note that on a Level 3 diagram (not shown), activities like **B.6.6**, **B.6.7** and **B.6.9** would be involved in further separation in order to arrive at the very basics of the system logic. Similar hand-sketched diagrams were made for all activities that required progressive levels of detail.

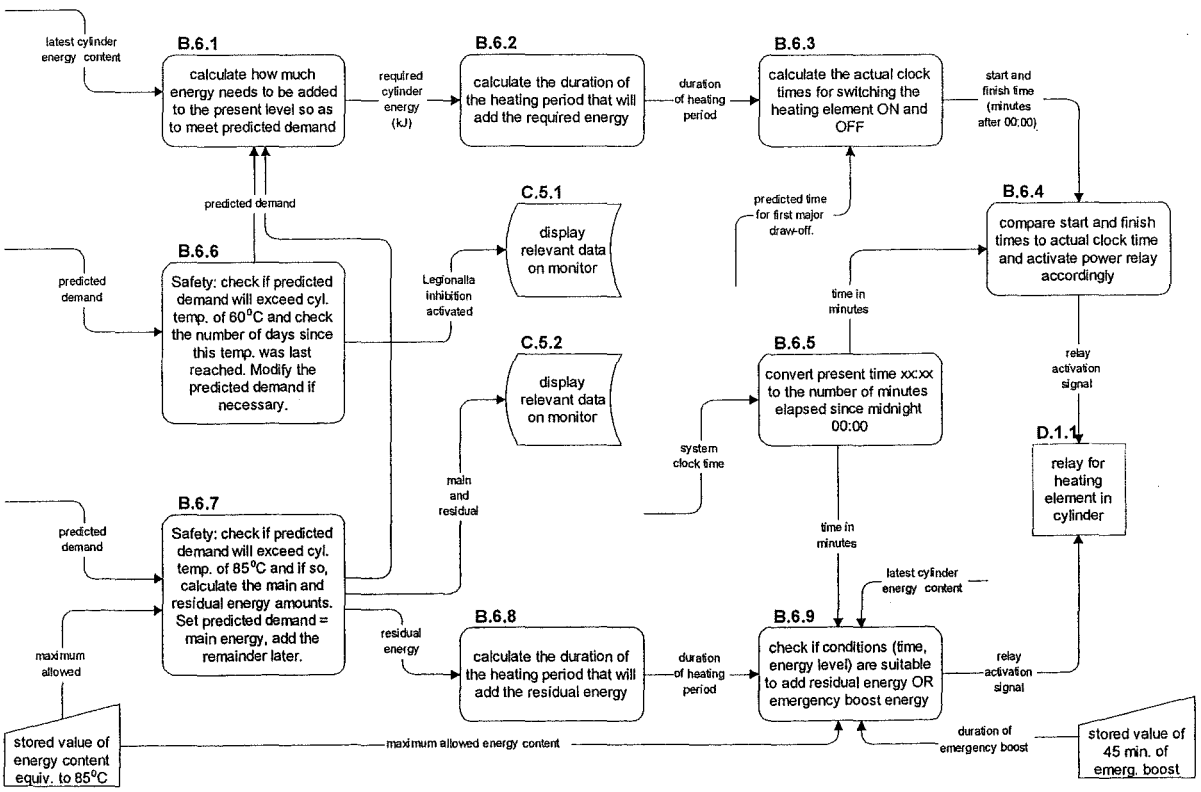


Figure 8.12 – **Level 2 data flow diagram for activity B.6 - FEMS**

Although a FEMS software *structure chart* was also mapped out, it has not been added to the thesis, being a detailed drawing spread over several sheets of A1 sized paper. The structure chart is a hierarchical, tree-shaped diagram and presents the next step in creating the structure of the software. It uses the previously designed data flow diagrams as a basis. Basically, the idea is to partition the data-flow diagram into its major *incoming* data, *outgoing* data, and *transform* data portions. Then each of these portions is partitioned into similar incoming, outgoing and transform data portions. In turn, these portions are subsequently partitioned, and so on, in a recursive manner, until the ultimate input and output sources have been reached. Each of these passes over the partitions is called a *factoring* of the system (Mynatt, 1990) and results in roughly one level or generation of modules being added to the structure chart representing the software structure of the system. Mynatt refers to this process as *Transform Analysis*.

8.4.3 Safety, diagnostic and optional features

Working with software and being able to relatively easily monitor and control the energy of any fluid system, allows the designer a great deal of scope in adding a number of interesting features to a system, which at any other time might prove too expensive, complicated or simply inopportune to add.

Safety is of importance in any engineered system that has some interaction with its users. Safety, aside from the power supply aspects, for a hot water cylinder in a conventional set-up usually limits itself to adding a tempering valve to avoid users burning themselves with high temperature water, and could also include adequately securing the cylinder for risk of an earthquake. Safeguarding against risk of disease, such as Legionella, can only be attained if the conventional temperature thermostat is set at a temperature in excess of 60 degrees Celsius. Under normal circumstances most domestically situated thermostats are set to a higher value, meaning that this might not prove to be a problem. It could however form a potential problem for FEMS.

This is because it is possible, in the case of a minimally used hot water system, to have FEMS arrive at the decision that the needs of the consumer are fully met by having a maximum stored water temperature of, say, 55^0C (for illustration purposes it is easier to talk in terms of temperature than stored energy). As such the consumer would run the risk, albeit small, of Legionella bacteria building up in the tank. The FEMS software overcomes this by monitoring whether the maximum temperature *at the bottom* of the cylinder remains below 60^0C for a period of 7 consecutive days. A software flag is raised when this situation is encountered, and subsequently the predicted energy value for the 8^{th} day consecutive day will be overridden if it is not equivalent to obtaining a (lower zone) water temperature greater than 60^0C. The override routine will set the energy level to give an average water temperature of 62^0C at the bottom of the cylinder, thus automatically ensuring that the rest of the contents will reach at least the same temperature. This is only valid for a single day; after that the system returns to the predicted state of energy use, and once again starts to count consecutive days in readiness to repeat the cycle should this prove necessary.

The risk of thermostat failure, although rare in occurrence and usually resulting in an open circuit condition, is greatly reduced with FEMS by having not one but a whole series of sensors. That is the advantage; the disadvantage is the increased risk of failure of at least one of the sensors at some point in the systems lifetime, or maybe even the complete sensor strip. Depending on the severity of the failure this carries with it a risk of the FEMS not being able to monitor an increase in cylinder energy and thus failing to switch of the element when the correct level is reached. A diagnostic module in the software tackles this issue by

continuously comparing the length of time the element is on with a pre-set maximum time value. If this value is exceeded than the program automatically switches the element off and raises an alarm flag on the display. Maintenance will then have to be carried out.

The consumer is given the option to indicate to the FEMS that the household does not want to run out of hot water at any time of the day. If this 'Emergency Boost' option is utilised, the system will act to heat the water for approximately 40 minutes should the energy level of the hot water in the tank fall below a pre-set minimum (equivalent to an average water temperature of 35^0C). The 40-minute heating interval should raise the water temperature to 45^0C in a 180-litre cylinder fitted with a 3kW element. Sufficient for most applications such as a washing machine cycle, a bath or a shower, and hopefully sufficiently short to avoid user annoyance. Under these circumstances the program will keep track of the number of times this heating occurred and add the additional energy required to the E_{total} value for the 24-hour period; thus aiming to avoid this shortfall in the future.

There exists a possibility that the predicted (and actual) energy demand is a value so high that, if applied, it would result in a cylinder full of water at a temperature greater than 85^0C. Theoretically boiling water would be possible, as the set point of the cylinder's temperature relief valve is usually 99^0C. Most domestic cylinders are not meant to continuously operate at temperatures exceeding 85^0C, and in fact the upper limit of most of the conventional thermostats is 80^0C. Yet, the energy demand might be valid, and, short of upgrading to a larger capacity cylinder, a conventional system would not be able to meet this high demand.

FEMS, however, is capable of this and does so by splitting the total (predicted) demand into a 'main' and 'residual' energy portion. The main portion is added as per normal, i.e. before the end of the night-rate at 7am, and does not let the water temperature exceed 85^0C. The resultant residual energy portion is added when the temperature drops below 80^0C. Should the cylinder temperature once again reach 85^0C, and do so before the complete amount of residual energy has been added, than the software will repeat the input cycle when the conditions are once again favourable. Only when the local time has gone past 10pm will the attempts to add the residual energy cease, and instead the program notes the actual amount added.

In an advanced system, such as the one presented in *Chapter* 9, the guidelines for adding residual energy can be altered to have the times of addition coincide with the periods of the lower tariff rates, as they occur throughout the day.

8.4.4 The user display – Dos and Windows

Any system designed for human use has two facets: functional capabilities and a *user interface*. The functional capabilities are the operations that the system is capable of carrying out. The user interface is the access the user has to those capabilities and usually takes the form of a monitor with keyboard/mouse input. Screen layout and design are an important part of most interactive systems. For FEMS the interface was designed very much as a separate system, which in terms of design is a definite advantage as it is generally considered that this leads to a less complex solution. It is also designed to be less of an *interactive* system and fulfils more of a *monitoring* role. Although it does not add to an increased performance of the FEMS algorithm it is easier for the human operator, whether in test mode or operational mode, to monitor the execution of the various tasks and the subsequent results if an informative screen display of the system operation is available.

The initial user interface purely displayed information of the various tasks that the FEMS was monitoring or performing and, having been written in the software language C, operated

under DOS on the PC. The display of operational parameters in real time previously consisted of simply outputting results as lines of cursor text, with no way of recalling 'on-line' results that had already moved off the screen. A stationary display, although not essential, was deemed advantageous for demonstration purposes. In a DOS form this proved to be more of a programming trial than initially thought, in the main due to the limited access to the DOS system (as made available in the C command structure). *Figure 8.13* shows the end result. The display allows no form of interaction and is solely designed to display the various parameters and results.

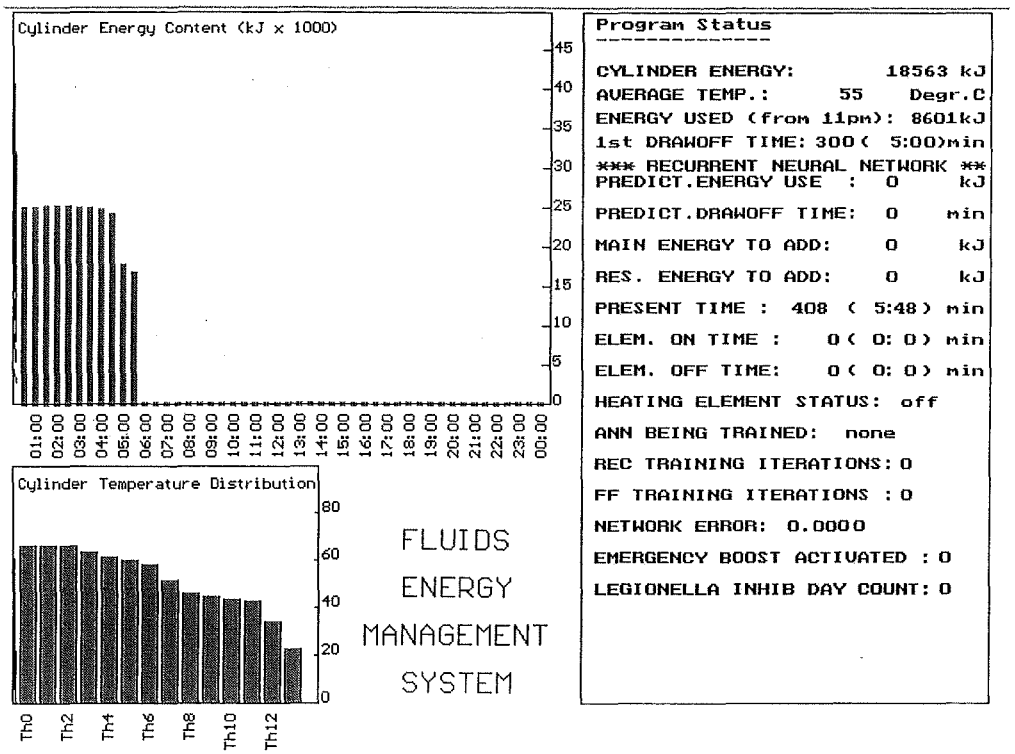


Figure 8.13 – **The DOS version of the user display (initial test phase of FEMS).**

In the top left-hand corner there is a bar graph showing the energy content of the cylinder from midnight onwards. A bar is drawn every half-hour on the hour. The present status of the water temperature in the various cylinder regions is shown in the graph in the bottom left-hand corner. The temperature readout is a straight conversion of that sensed by the thermistor in each respective region. Th0 represents the top of the cylinder and stands for 'thermistor no.0'; being the first thermistor sensor on the sensing strip placed along the length of the cylinder. The block on the right-hand-side shows a number of important parameters such as the amount of energy presently in the cylinder and the average temperature of the water. Time for various parameters or program actions is monitored and displayed in terms of the minutes elapsed from 11pm onwards as well as the standard format of hh:mm. Although not in use when this snapshot was made of the display, also shown are the predicted values for first draw-off and total energy usage as derived from the two neural networks in conjunction with the 'network error' (Mean Relative Error). At the bottom of the column it is possible to read of the instances of the emergency boost having been used, and the number of days that the cylinder temperature was low enough to allow the possible build-up of harmful bacteria.

The Windows version that followed on the rather austere Dos version was programmed using Visual Basic (VB3.0), and expressed the desire to be able to display a larger range of parameters in different formats, specifically graphs, in real time. Visual Basic gave added design flexibility in addition to a more professional presentation style. It did mean however that the existing FEMS program needed modification to run under Microsoft Windows, elimination of the 'system' calls being the first and obvious alteration for those familiar with 'C'. A larger number of graphs can also be displayed and this required extra modules to be added to the existing software. It should be noted that the author had not programmed previously with VB 3.0 and there is undoubtedly room for improvement in the design and function of the Windows user interface.

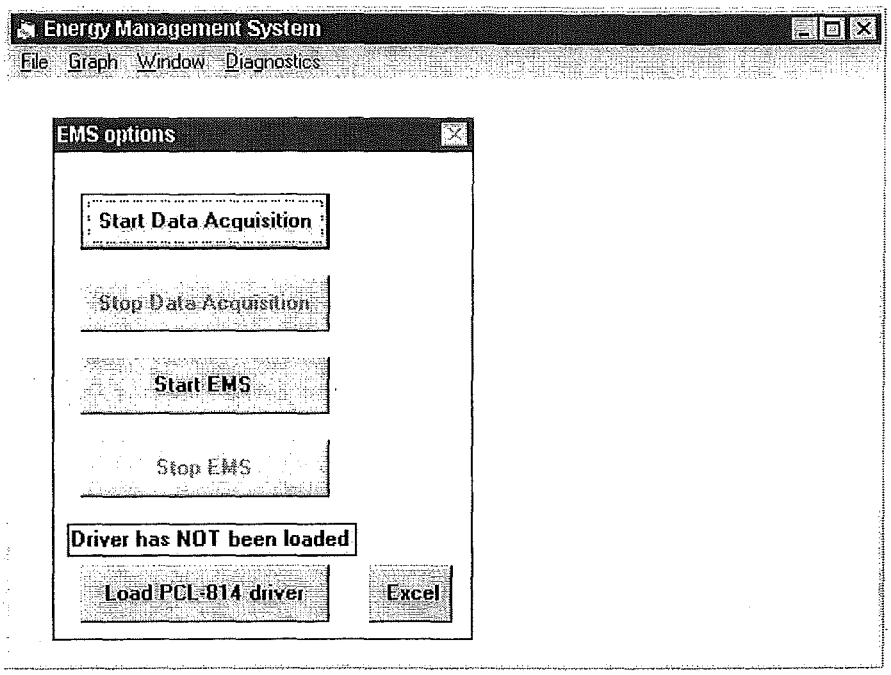


Figure 8.14 – **The main window for the FEMS user display clearly illustrates that the program is controlled via the interface buttons in contrast to the Dos display.**

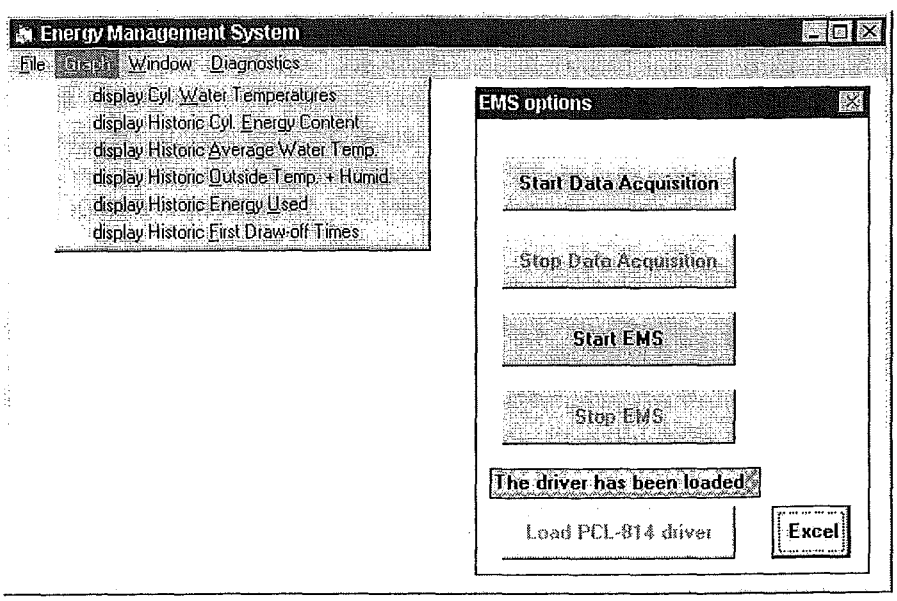


Figure 8.15 – **different graph options are accessible via a pull-down menu.**

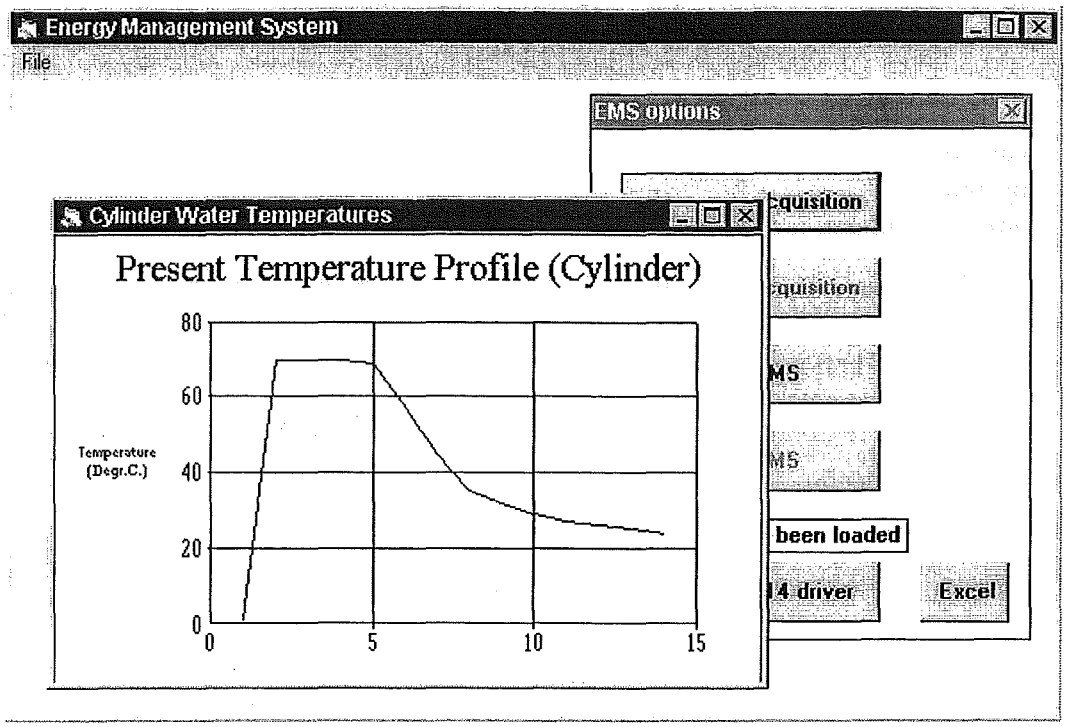


Figure 8.16 – **The graph that is shown when the user clicks on the "display Cyl. Water Temperatures" option on the "graph" pull-down menu.**

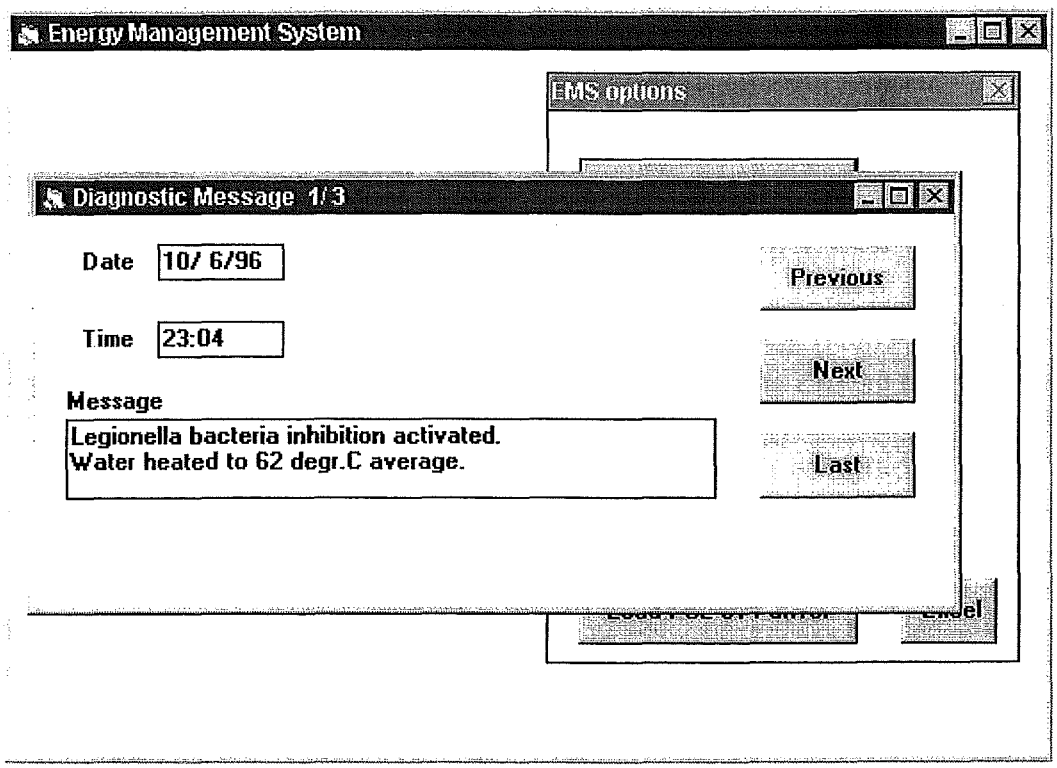


Figure 8.17 – **System diagnostic messages can be displayed by clicking on the "diagnostic" pull-down menu. Available messages can be cycled through by using the three buttons on the RHS.**

The major difference between the Dos and Windows versions lies in the fact that the latter has a truly interactive display. In fact the FEMS program is started, actively monitored and terminated via the interface (see *Figure 8.14*), whereas the Dos interface only allowed passive monitoring. *Figure 8.15* to *Figure 8.17* show examples of what is available for selection and display by using the menu options. The graphs are of a simple format and need enhancing, but for the purpose of the thesis their main aim is to show what is possible in terms of information and feedback.

A last mention should be made of the fact that the starter window as shown in *Figure 8.14* allows the user to have the system software either collect hot water usage data only ('Start Data Acquisition' button) or launch the full energy management program with the data collection and neural network predictions ('Start EMS' button).

8.4.5 Software summary

Version 1.0 of the FEMS as completed has the following features:

- Consists of approximately 4000 lines of code distributed over 25 modules.
- Has an executable file size of 121 Kb and utilises the Large Model with far pointers.
- Uses a real-time kernel to ensure prioritised event handling.
- Has been tested for module integration and system integrity using simulation code and fictitious data.
- Includes a recurrent and back-propagation form of the neural network.
- Stores up to 12 weeks (84 days) of historic data for neural network learning.
- Features Emergency Boost, Legionella Inhibition, Main and Residual Energy Transfer, Safety Energy Limitation, First draw-off determination and Energy/Draw-off time forecasting.
- Provides a Dos-graphics display showing the energy history, cylinder temperature distribution and software status messages or a Windows user interface with similar and additional features in terms of control and information exhibited.
- Written largely in ANSI C to ease transition to a small dedicated microprocessor; to this end the structure of the software has also been kept simple. The code size should reduce further when the Dos and/or Windows overhead is eliminated from the software.

8.5 Establishing the neural network design

FEMS aims to achieve its working objectives by making a reliable prediction of both daily hot water demand and the time of the first significant draw-off. In fact, its prediction capabilities form the crux of the system; without this there is little value in having a fluid energy management system installed in the first place. At a certain stage in the project it appeared that *accurately* predicting the hot water use of a single family was substantially more difficult than forecasting, say, the next 24 hour load of an electricity supply network (and based on the number of papers on load prediction, this is obviously deemed hard enough in itself from the viewpoint of the supply authorities). The greatest contributing factors to this additional hurdle for the FEMS is that there is a lack of a large population acting as an averaging or buffering mechanism, and that there is little data for household hot water demand. Statisticians always favour a large amount of information when making predictions of any sort, and with good reason, as it negates the effect of the 'outliers' in a

data series and tends to give a normal distribution to any data gathered. Unfortunately this mitigating factor totally disappears when working with just a single-family unit.

This disadvantage can be offset if a large amount of data is collected from each domestic dwelling where a FEMS is installed prior to the system becoming fully operational. But it is FEMS stated aim, once installed, to be up and running and making reliable forecasts with a minimum amount of delay. This entails that only a small amount of historic data will be available for neural network learning (thus also saving on memory storage space), and this raises the obvious question of what the minimum amount of information needs to be to make reliable predictions. It also means taking a prediction model to its design limits by means of fine-tuning the data quantity, the data input and the processing power.

(Larger fluid demand systems, as can be found for instance in industry, should have less of these negative factors and could thus be blessed with a more predictable behaviour. Alternatively there are neural networks that specialise in predicting chaotic time-series; this is a complete branch of research in itself and offers further possible scope for the FEMS future).

As mentioned above, one of the main objectives of FEMS is to achieve a realistic prediction of both daily hot water demand and the time of the first significant draw-off. The intention is to make these forecasts by means of Elman recurrent neural networks. A number of network design parameters need to be determined by means of the tests that can be made on the available historic data and the software module for Elman network. These parameters are optimised from the standpoint of improving the prediction accuracy, and can be summarised as follows:

- Determine the amount and type of historic data necessary.
- Find the optimum neural network configuration in terms of the number of input and recurrent neurons (the output layer is fixed as a single neuron).
- Establish the number of training epochs (an epoch is single pass through all the training input and target vectors) and the learning error.

The first point with regards to the *type* of historic data that would prove optimal needs some clarification. The fact that the delay in a recurrent network facilitates the storage of values related to the previous time steps means that it should be possible to present as input (to the neural net) not a whole window of past values but a single value instead. Thus the training vector as shown in *Figure 8.7* and *Figure 8.8* could be reduced to having 'day 1' as an input, plus usual temperature and Boolean values for day, month, etc. The advantage is obvious: less inputs means less processing and thus eases the load on the microprocessor and the working memory.

The neural network will therefore also be tested with various combinations of input data; specifically:

- day 1 on its own,
- day 1 plus the 3 Boolean values for day,
- day 1 plus all (8) of the Boolean values,
- day 1 plus temperature and all (8) of the Boolean values,
- day 1 to 14 on its own,
- day 1 to 14 plus the 3 Boolean values for day,

- day 1 to 14 plus all (8) of the Boolean values
- day 1 to 14 plus temperature and all (8) of the Boolean values.

This should give a clear idea of which data makes a significant contribution to the prediction process.

8.5.1 Training the neural network

From the previous chapters on neural networks and their applications it is clear that a useful network, 'useful' in terms of outputting valid information, can only be arrived at by entering valid and appropriate input data; and the more the better. Thus, the neural network needs to undergo a proper training or learning session before being actively engaged in prediction.

Hot water usage data, collected from the attic installed cylinder in the author's house over a period of about 6 months gave the requisite figures and allow a body of training data and a body of testing/validation data to be formed. After training with the first set, the prediction model is tested on the next day. 24 hours later the window is advanced by a single day and this process is repeated (*Figure 8.18*).

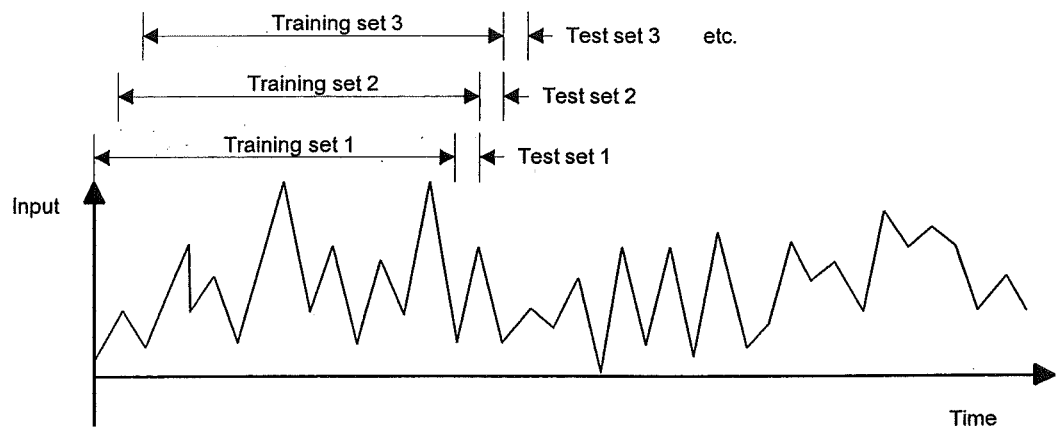


***Figure 8.18* - A depiction of the training and test sets used.**

A concession was made to the ease of being able to directly interpret the results of the predictions. The daily amount of hot water energy used in the household was replaced in each energy-training vector with its equivalent quantity in litres. Both quantities had been logged by the data collection system. As mentioned, the motivation for this is that it is easier to comprehend the smaller figures and the measuring unit when working with the familiar *litres* of hot water used c.f. that of *kilo-Joules* of energy.

For instance, if the results state that the RMS (root-mean-square) error is 9 litres for a particular set of predictions, than this is a quantity that relates in an intuitive manner to the total cylinder volume of 180 litres and it needs no further translation. Alternatively, an RMS error of 1592 kJ would need to be expressed as, for example, a percentage of the total available energy in order for the value to provide some form of insight.

8.5.2 Results

As mentioned in *section* 8.3.3, the first prediction is made when 4 weeks worth of historical data has been formed into 14 training vectors; each vector containing 14 points of contiguous historical data, plus the relevant values for temperature, day of the week, month, etc. The accuracy of this prediction can be determined 24 hours later when the actual value becomes

available and is added as the latest point to the historical time series and the next training vector.

The size of the training set, the number of neurons in the hidden (recurrent) layer and the number and type of values that build-up an individual training vector, were the three parameters that were altered in fixed steps during the trials. The values that each parameter can take, and the number of possible combinations tested, are as follows:

PARAMETER	POSSIBLE VALUES
The number of vectors in a training set :	14 28 56 84
The number of neurons in the hidden layer :	5 10 15 20 25 30
The sort and amount of data in each vector :	ranging from *day1 only* to the full *14 days + temperature + 8 Boolean values*

Given the large number of results derived, only those that show potential and those with interesting possibilities or ramifications will merit further discussion.

The only other parameter that needs explaining is the training error, which is the sum-squared error goal that the neural network tries to reach when training with the available vectors. This value was fixed at 0.01. However, regardless of the fact that the error goal might not have been achieved, the training terminated automatically after a set number of epochs. Initially this was set at 500, but observance of the network error during training quickly showed that continuing the learning past 200 epochs returned little in the way of further significant error reduction. Limiting the number of epochs to 200 gave other advantages that could not be ignored; it reduced possible overfitting and cut the training time by more than half.

8.5.2.1 Hot water demand predictions

The results for the most challenging time series will be discussed first. *Figure 8.19* shows a plot of the *hot water demand* over a period of 144 days, from July to December.

The graph is interesting in that it shows a trend towards less hot water use in the warmer months of November and December. The explanation here is that one adult member of the farming family was not present from the end of October onwards. The rise in demand over the month of September is less easily explained. It could possibly be linked to additional outside activities and a corresponding increase in shower usage. All in all it is considered a very suitable set of data for testing the adaptive response of the recurrent neural network to changing environments.

The shape of the distribution curve in *Figure 8.20* reinforces the fact that this time series presents a challenging set of data with a possible chaotic component. The spread of the data is considerable and the characteristic single bell shape of a normal distribution is absent. Rather there is a skewed bell shape centred on 40 litres and a smaller peak at 130 litres of hot water draw-off.

Subsequent to the tests it became clear that a number of the Elman recurrent neural network combinations give good results, but only one version stands out and returns a superior performance. This preferred version is a network fitted with 20 recurrent neurons in the hidden layer and 9 nodes for the input layer. The reason for only having 9 values in the input vector is that the best result is reached with a training vector containing just one historic data value (day 1) and all the 8 Boolean values for day, month, holiday and special day.

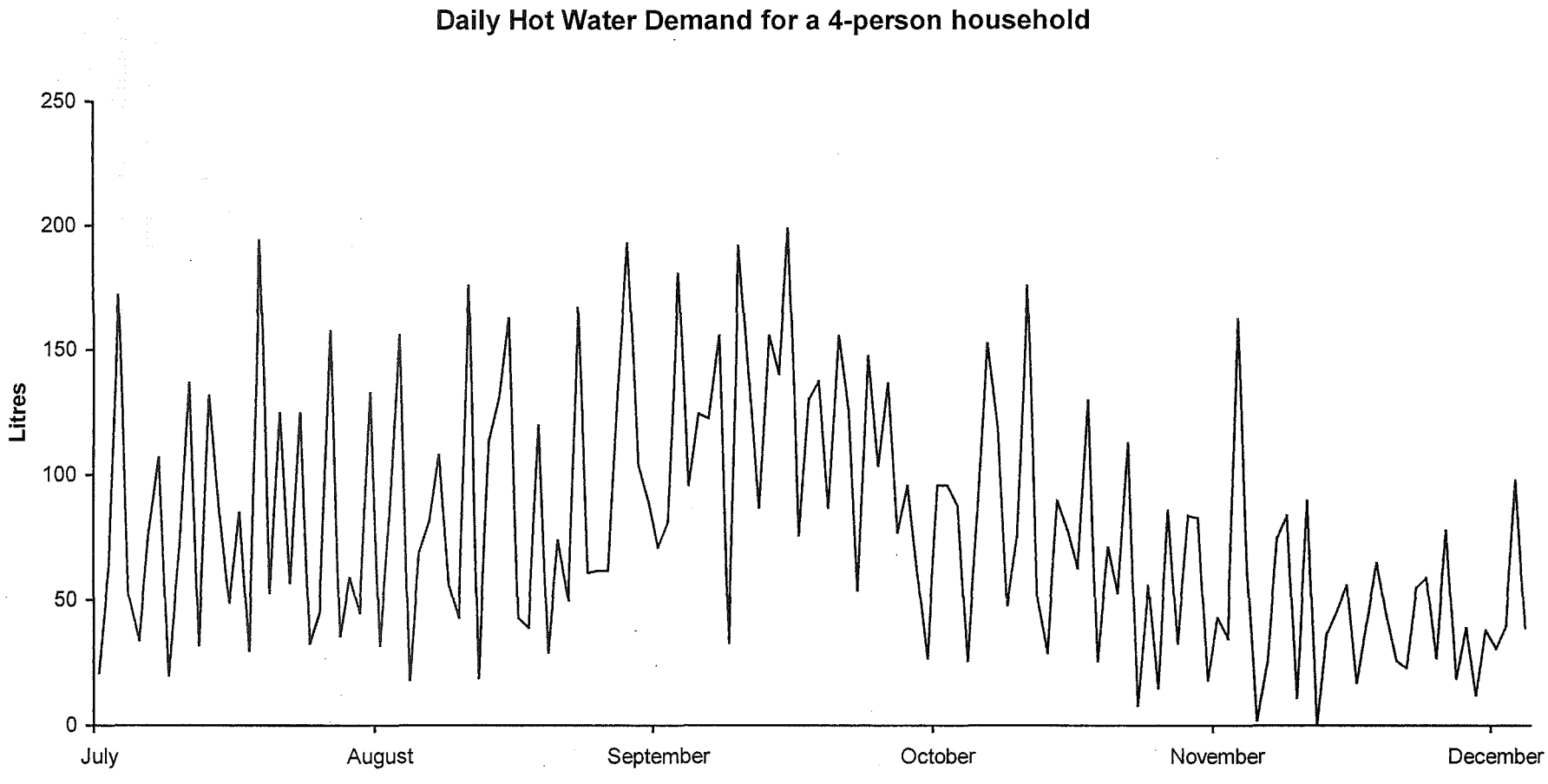


Figure 8.19 – **Actual daily hot water use over a 6 month period for a 2 adult + 2 children family**

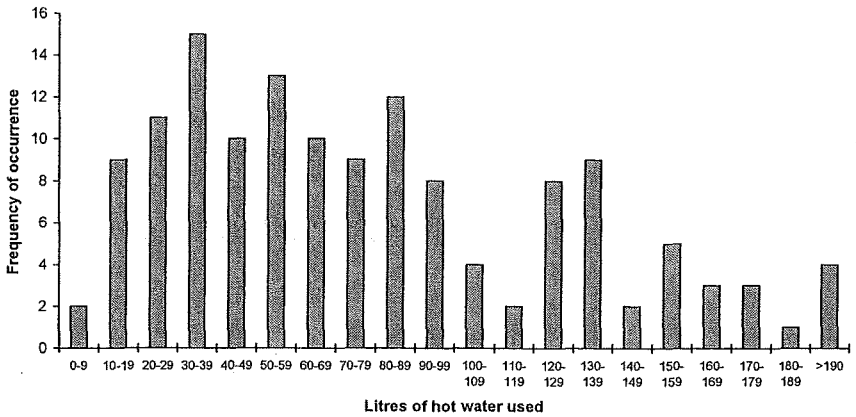


Figure 8.20 - **Discrete distribution curve for the daily hot water demand**

The temperature values did not appear to contribute in any significant manner in the better results and for all intents and purposes can be eliminated from the FEMS training vectors. The results also give us the preferred amount of historic data needed for training the network: 56 training vectors, representing 8 weeks of in situ collected data, span the length of the moving window.

Examination of the actual time series versus the predictions, as shown in *Figure 8.21* and the associated error (*Figure 8.22*) illustrates that the neural network almost immediately finds the correct range of the demand values and begins to match the trend, if not quite the absolute values, of the actual time series. It loses this synchronicity after only 19 days in the prediction phase however, and the network goes through an extra learning process before the predicted values recover and start tracking the actual demand data. This extra recovery/learning time of approximately 18 days comes in addition to the daily training with the 56 vectors from the advancing window. After 37 days from when forecasting was started the predicted values are once again in step with the daily volume of hot water that is really used. The RMS prediction error for the last 3 weeks is 13 litres, or 7% of the total volume of 180 litres; this can be considered a promising outcome.

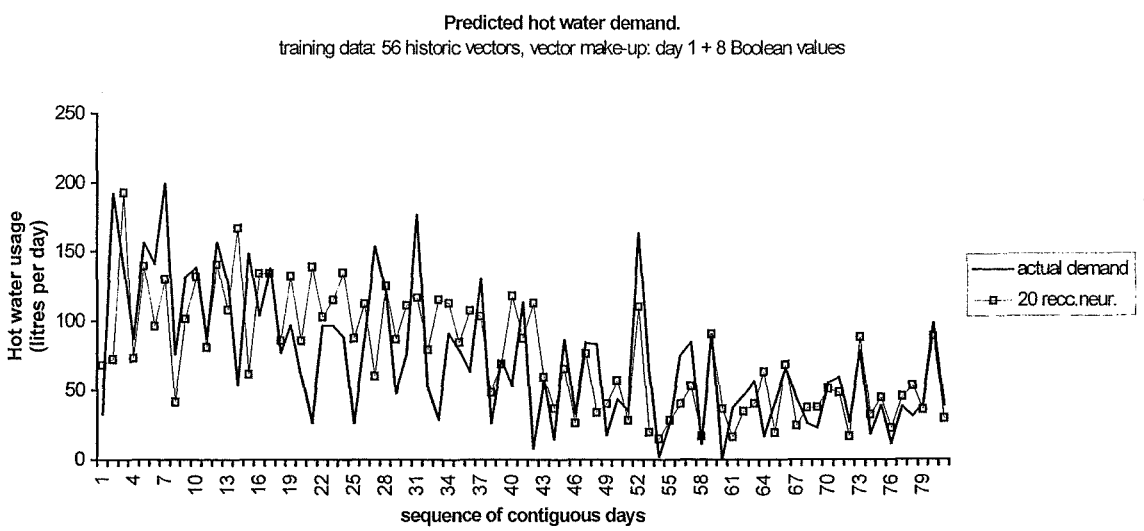


Figure 8.21 – **A graph showing the actual demand and the predicted demand made with a recurrent neural network with 9 input nodes and 20 recurrent neurons.**

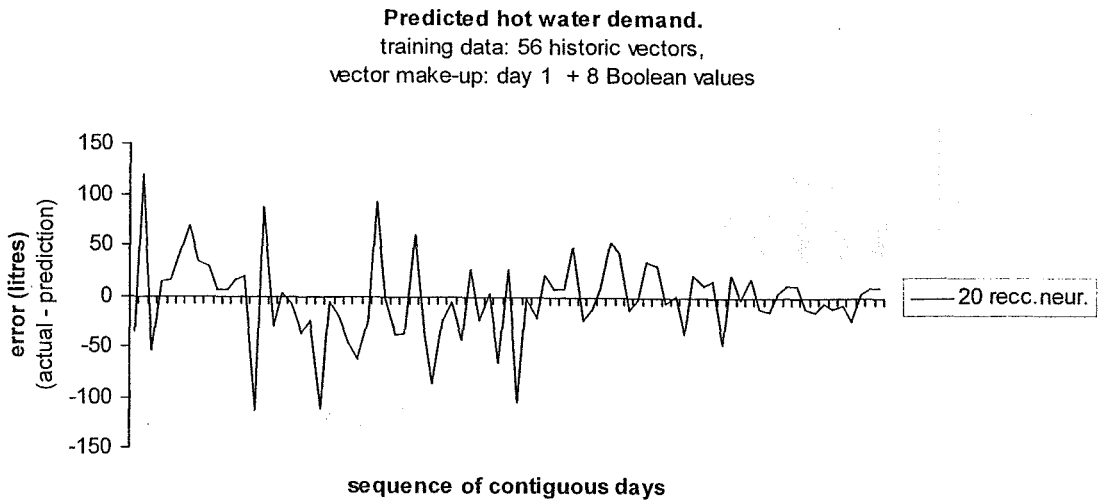


Figure 8.22 – **A plot of the difference between actual and predicted values, trending towards zero as the prediction accuracy improves.**

Before examining the results more closely a note of caution. Although the error results are shown as continuous curves from the viewpoint of clarity, it should not be forgotten that these are *discrete* values. For recurrent neural networks it is not wise to interpolate/extrapolate the individual result. It is possible, for example, that a 7 recurrent neuron network will provide a much better or worse outcome than the adjacent 5 or 10 neuron versions, it cannot be safely assumed that it will lie between the two.

A look at *Figure 8.23* provides more information with regards to how well the different combinations of recurrent neurons with the different types of input fare when faced with a learning set of 56 training vectors. The three curves show the root-mean-square error for the predictions made in latter part of the hot-water demand data string; specifically the last three weeks when the predictions match the actual values closely. The *day 1 + 8 Boolean values* type of input stands out as providing the smallest RMS error result regardless of the number of recurrent neurons in the hidden layer. As already discussed, 20 neurons gives the lowest value error of 13 litres. But if some of this accuracy is sacrificed than having only 5 neurons in the recurrent layer, with an RMS error of 16 litres, would allow a notable decrease in the number of network weights and hence the accompanying calculation and memory storage. Whether this trade-off is worthwhile will depend to a great extent on the type of microprocessor/memory size that is eventually utilised for a commercial version of FEMS. What should be kept in mind is that the ability of the neural network to cope with more demanding patterns is significantly curtailed with fewer recurrent neural units

Unexpected is the word that best describes the other two curves in *Figure 8.23*. Based on what has been said so far with regards to the added value of having the Boolean inputs, it would seem logical for the *day 1 to 14 + 8 Boolean* values to provide a lesser error than that of just the 14 past values of the demand series itself. But this is clearly not the case and suggests that the neural network is unable to sort out and utilise the extra information contained in the larger of the training vectors in any decisive and advantageous manner. That this type of vector should not be dismissed out of hand though is reflected in the results achieved when a larger set of 84 training vectors (12 weeks of historic data) is applied to the learning of the network, as can be seen in the next graph, *Figure 8.24*.

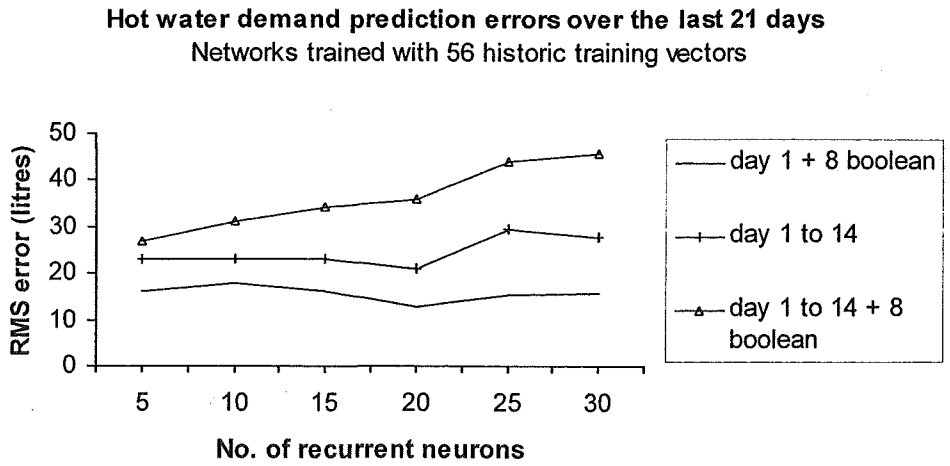


Figure 8.23 – **A display of the effect the three different types of data input have on the RMS prediction error over a 21 day period and trained with a set of 56 vectors.**

Although in a general manner the *day 1 + 8 Boolean* data vector still provides the least amount of RMS error just as for the 56-vector training set, on a case by case basis the 5 and 25 recurrent neuron configurations result in the lowest errors while learning with a *day 1 to 14 + 8 Boolean* data vector. This points to the neural network needing a larger learning database before it can make use of the additional information in the largest of the training vectors. A plot of the actual and predicted hot water demand using the 5 recurrent neuron network and this large vector is seen in *Figure 8.25*. The RMS error in this case is 19 litres over the last 3 weeks of data, c.f. 16 litres with a 5 neuron, 56 vector, *day 1 + 8 Boolean values* version of the neural network.

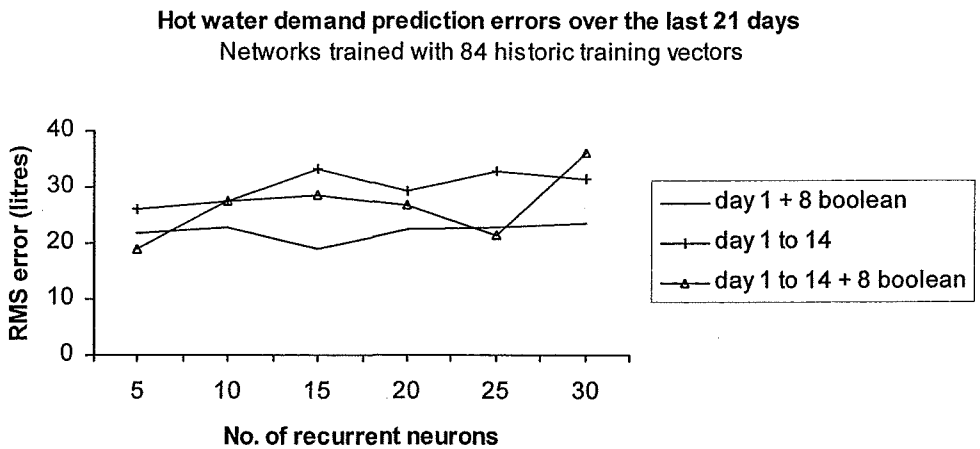


Figure 8.24 - **A display of the effect the three different types of data input have on the RMS prediction error over a 21 day period and trained with a set of 84 vectors.**

Before moving on to the First draw-off time results, a final analysis will be made of the influence of the size of the training set. Predictions were made using 14, 28, 56, and 84 training vectors in a set. From the previous paragraphs it has already been revealed that 56 and 84 vectors give good working results. The three graphs in *Figure 8.26* illustrate that same conclusion once again for three of the eight available types of training vector tested.

For *day 1 to 14 + 8 Boolean* data vectors the worst errors are made with the smallest of the training sets, 14 vectors, and the most accurate figures are achieved with largest set of 84 vectors. This same assertion can be made for both the *day 1 to 14* and the *day 1 + 8 Boolean* data vectors, except that the best figures here are achieved with the training set of 56 vectors, and not 84. The general remark here must be however, that additional training data improves the prediction results.

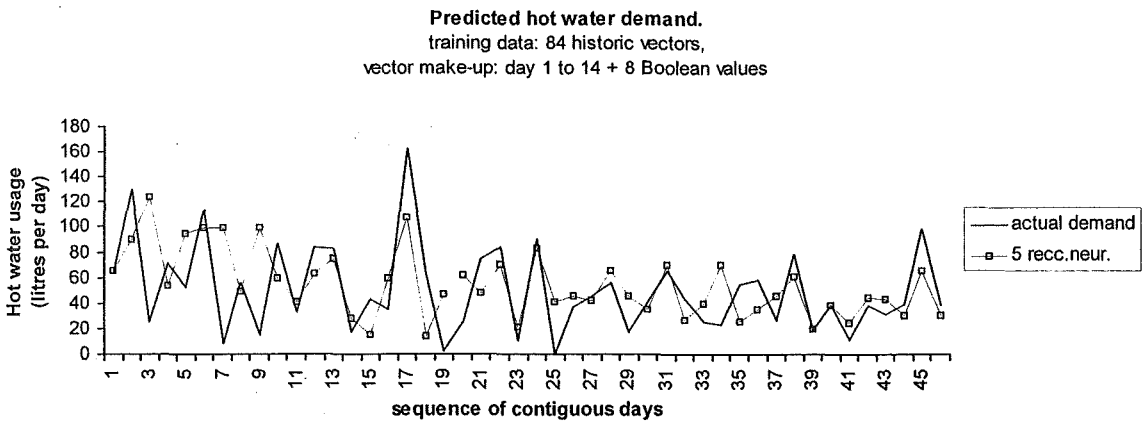


Figure 8.25 **- The actual demand and the result of predicted demand made with a recurrent neural network with 22 input nodes and 5 recurrent neurons**

8.5.2.2 First draw-off time predictions

The treatment given to the data series formed by the daily First draw-off time was exactly the same as that of the daily hot water demand. Different combinations of data input, recurrent neurons and training set sizes were let loose on the data as shown in *Figure 8.29* and the resultant predictions were scrutinised for error and fit.

A comparison of this time series with the one for hot water demand reveals a less tumultuous, more regular characteristic, compounded by sudden peaks that the neural network might have trouble tracking on a short-term basis. In the longer term these peaks would be interpreted as noise and once learnt should help to make for a more robust network.

The combination of data and neural network configuration that dealt best with this time series was a 30 recurrent neuron version trained with 56 historic vectors, each vector being of the type *1 to 14 days + 8 Boolean*. Again, as for the hot water demand, the temperature data does not figure in this training vector having shown no positive influence on the results. *Figure 8.27* presents the prediction outcome over a period of 75 days.

As for the prediction curve of the hot water demand it is apparent that an additional learning period is necessary for the predictions to run synchronised with the actual data.

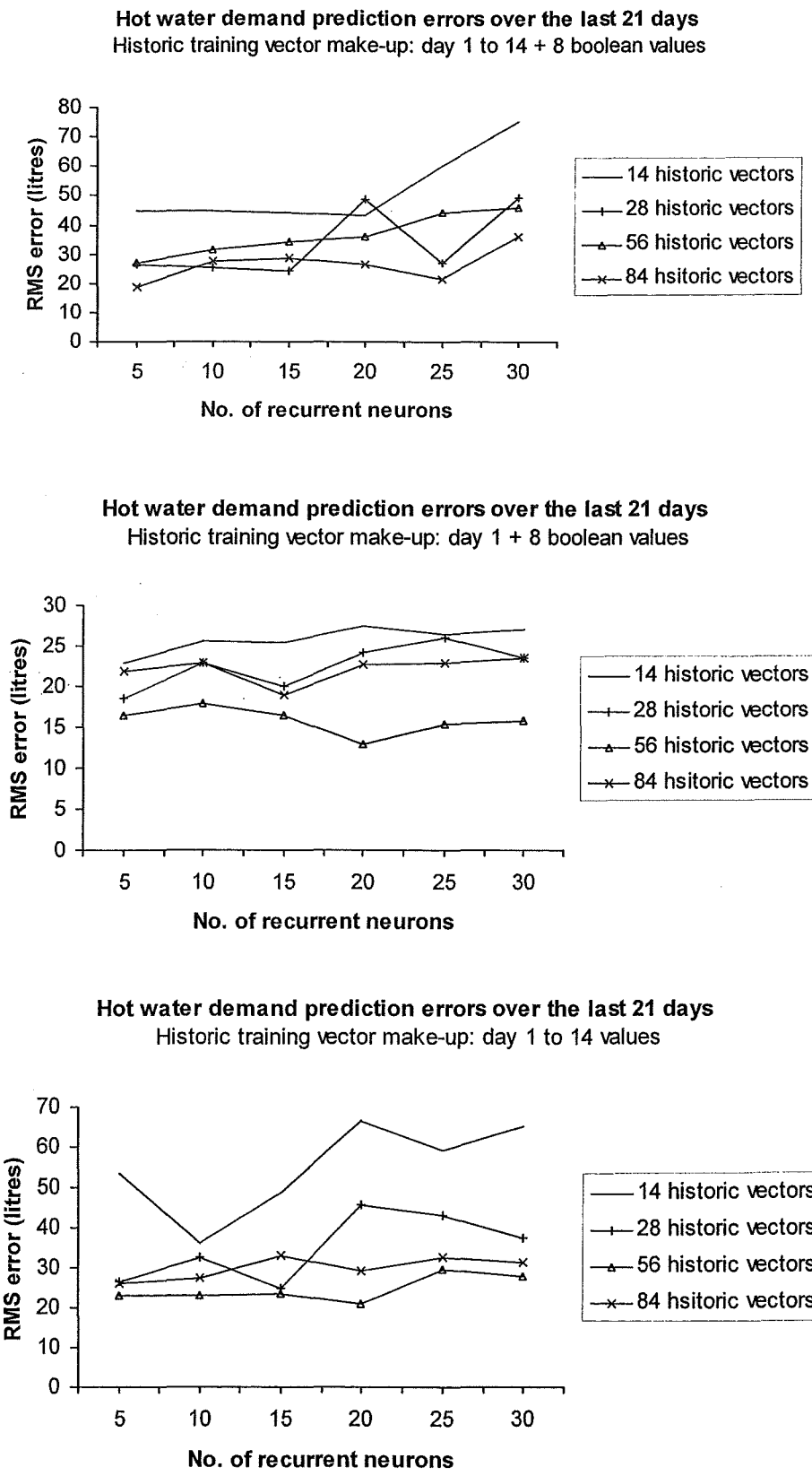


Figure 8.26 –**Three graphs showing the influence of the size of the training set (14, 28, 56 and 84 vectors) on the prediction error. Each graph concentrates on a different data vector:** *day 1 to 14 + 8 Boolean*, *day 1 + 8 Boolean* **and** *day 1 to 14*.

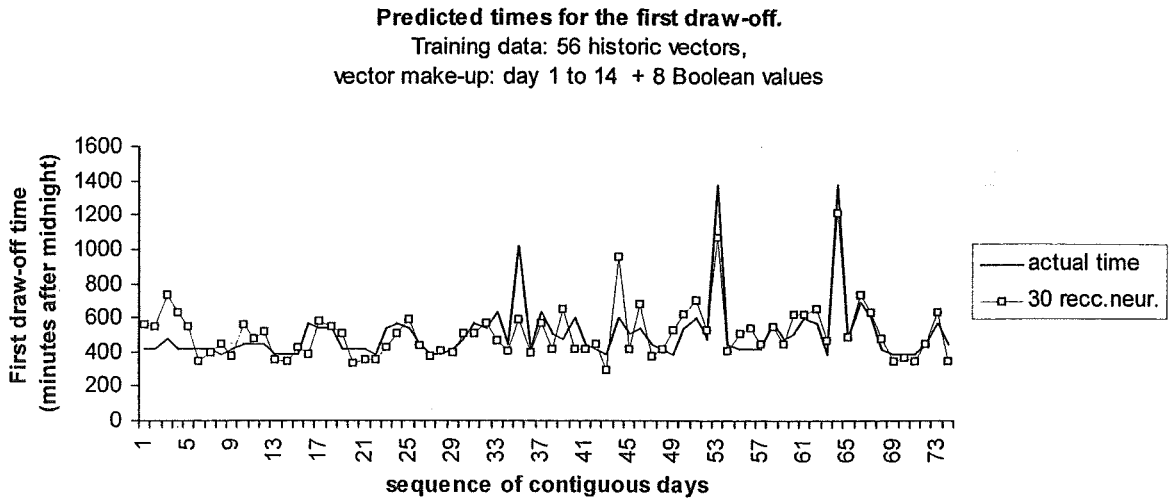


Figure 8.27 -**The actual draw-off time and the result of the predicted time made with a recurrent neural network with 22 input nodes and 30 recurrent neurons.**

For *Figure 8.27* this is achieved after approximately 26 days into the prediction sequence. Not surprisingly it is unable to forecast the sudden late start in hot water use on day 36, but tries to compensate a week later only to find that the draw-off time has now returned to a more average figure. The later peaks do get matched though, although not in the exact magnitude, indicating that the neural network has learned well in the intervening period.

There is some evidence also that use is made of the information provided by the Boolean values for day, month, etc., certainly with 30 recurrent neurons, but the error results using only the values for *day 1 to 14* are still remarkably close, as is apparent from *Figure 8.28.*

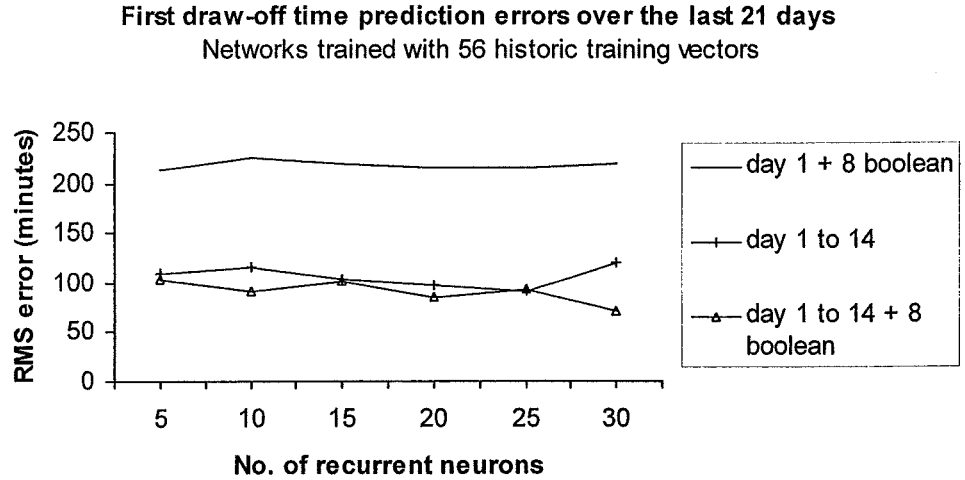


Figure 8.28 - **A display of the effect the three different types of data input have on the RMS prediction error over a 21 day period and trained with a set of 56 vectors.**

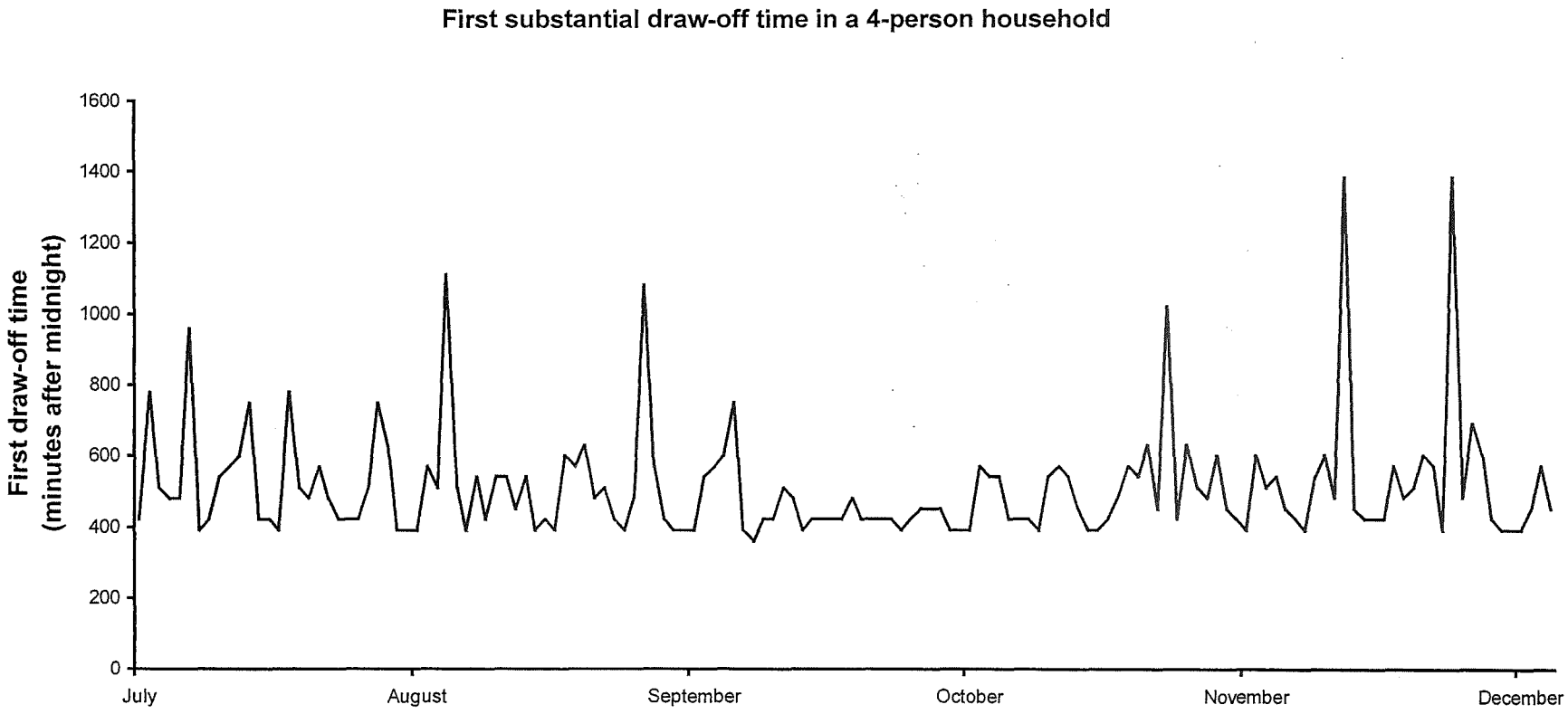
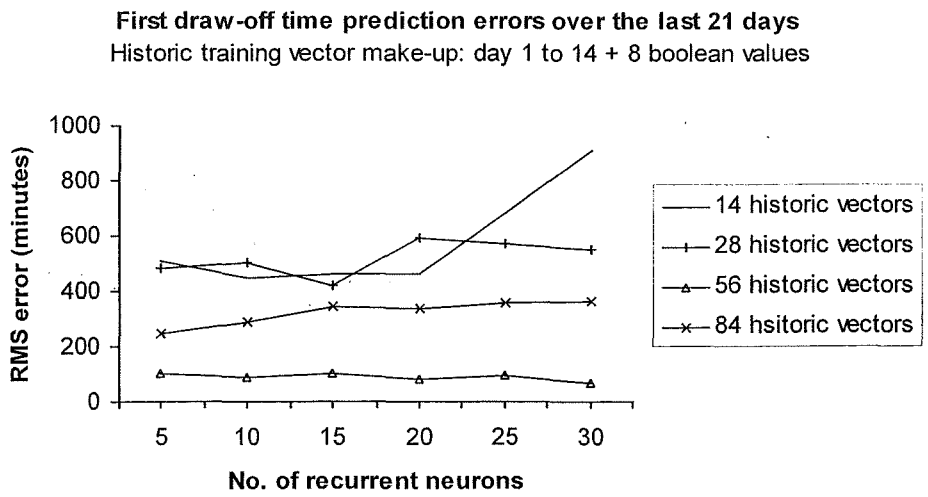


Figure 8.29 – **A time series plot of the actual time of the first substantial draw-off occurring (once-daily) over a 6 month period for a 2 adult + 2 children family.**

Whereas previously with hot water demand the *day 1 + 8 Boolean* data type was a big winner, here it fails to get within the proximity of the results that the other two data types provide. Clearly the neural network requires the information contained in the longer training vector of 14 days.

The RMS error for the 30 recurrent neuron case is 71 minutes, or just over an hour. As a percentage relative to 24 hours this is equal to 4.9%. The average error for the other neuron combinations hovers around the 105-minute mark giving an error percentage of 7.3%.

The effect of the number of training vectors utilised for building up the model in the neural network is displayed in *Figure 8.30*. The training set incorporating 56 vectors reveals a definite prediction error advantage over the other three quantities of 14, 28 and 84 vectors; this is irrespective of whether a *day 1 to 14* or the *day 1 to 14 + 8 Boolean* type of vector is used in the daily learning exercise.



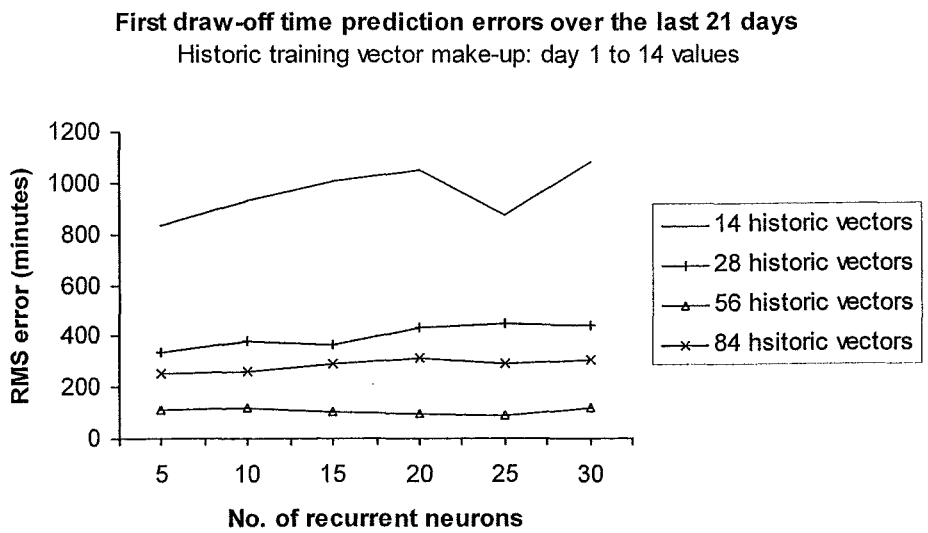


Figure 8.30 **–Two graphs showing the influence of the size of the training set (14, 28, 56 and 84 vectors) on the prediction error. Each graph concentrates on a different data vector: *day 1 to 14 + 8 Boolean* and *day 1 to 14*.**

8.6 Discussion

It was noted that the historic data used had all the hallmarks of a noisy, non-stationarity time series; with special emphasis on the series for hot water demand. Prediction is fundamentally difficult for such data because the problem of learning from examples is fundamentally ill-posed, i.e. there are infinitely many models which fit the training data well, but few of these generalise well. In order to form a more accurate model it is desirable to use as large a training set as possible. However, for the case of highly non-stationary data, increasing the size of the training set results in more data with statistics that are *less relevant* to the forming of a good prediction model. This might explain why a training-set size of 84 vectors returned a worse prediction performance than a lesser training set. In contrast, the high noise and small data-sets make the models prone to overfitting. Random correlations between the inputs and outputs can present great difficulty. The conventional models typically do not explicitly address the temporal relationship of the inputs, e.g. they do not distinguish between those correlations that occur in temporal order, and those that do not. This was a good reason for having chosen recurrent neural networks, as RNNs are biased towards learning patterns that occur in temporal order, i.e. they are less prone to learning random correlations that do not occur in temporal order. An extra measure taken to avoid overfitting with the small training data set was that learning was stopped early, 200 epochs being chosen as achieving a good network error (on average) as well as avoiding the overfit.

Training RNNs tends to be difficult with noisy data, with a tendency for long-term dependencies to be neglected (experiments reported in Lawrence et al. (1996) found a tendency for recurrent networks to take into account *short-term* dependencies but not *long-term* dependencies). There is a very real risk for the network to fall into a naive solution, such as always predicting the most common output. An attempt has been made to address this problem by providing additional information about the time span of the data, in the form of Boolean values for day and month.

8.7 Conclusion

The results are very encouraging. Despite the uncompromising characteristics of especially the hot water demand time series, it has been shown that good multiple time series predictions can be made with a small to moderate amount of historic information. Not surprisingly the prediction accuracy improves when more past data becomes available and is incorporated in the neural network weights and bias values during training; this supports the outcome of those publications that deal with the same topic, i.e. neural networks and time series prediction.

A Fluid Energy Management system fitted with Elman recurrent neural networks is capable of forecasting both the demand for hot water as well as the time of first use, for a family of 2 adults and 2 children in a conventional domestic dwelling. It is prudent in terms of minimising the error to collect at least 8 weeks of data, *in situ*, before commencing with predictions. The exact configuration of the neural networks might vary from domestic situation to situation, but a good starting point is 9 – 25 – 1 (9 input nodes – 25 hidden layer recurrent neurons – 1 output neuron) for Hot Water demand prediction and a 22 – 30 – 1 neuron configuration for First Draw-off Time prediction.

The predictions for hot water demand benefit from having the neural network trained on a daily basis with the 56 training vectors, each vector containing the last known demand value

plus the 8 Boolean digits which indicate the day, month, holiday and special day for that day for which the demand is being predicted.

For the draw-off time predictions each training vector should be made up out of the 14 last known time values plus the 8 Boolean digits which indicate the day, month, holiday and special day for that day for which the demand is being predicted.

Should the processing power and memory capacity of the (to be) employed microprocessor in the FEMS equipment prove limited than the number of hidden layer recurrent neurons can be reduced to 5, and the input vector to a basic set of 14 past data values. The trade-off being an increase of 3 to 5% in RMS prediction error.

8.8 Summary

Although this chapter by no means finalises the design and improvements to the FEMS, it does draw a line under the practical phases of the thesis. The current view of what FEMS should entail if is to be installed in a domestic or industrial environment on a short-term basis is expressed in the preceding chapters. FEMS as it stands exists as a combination of various bits of hardware and a number of software modules designed to run on a 486 type PC with 4 Megabytes of memory and a small harddrive. The next logical step would be to find a suitable microprocessor or DSP, scale the software down to run on the processor of choice, and then to gather more data on its performance by installing several systems in domestic and/or industrial facilities.

With a plausible working system as a foundation for further study, it is now time to devote a final chapter to what is feasible with FEMS if a dynamic, domestic consumer oriented, retail electricity market opens up in New Zealand or elsewhere. A competitive market where the small consumer is faced with fixed or varying pay rates of electricity, known as tariffs, throughout the day. Only an automated system could cope with the added complications of finding the more cost-effective periods of electricity and utilise this information in heating water or other mediums in an economic manner. *Chapter* 9 looks at the opportunities.

Chapter 9. Dynamic Tariffs and eFEMS

9.1 Introduction

It is not the intention of this last chapter to construe a detailed analysis of the latest developments in the energy market, be it on a global basis or just New Zealand on its own. Rather it intends to review briefly the concepts and explain what the pressures are on the power supply and distribution companies to introduce spot price tariffs, alternately referred to as 'spot price', 'spot tariff', or 'dynamic tariff'. The potential utilisation of various forms of tariffs, how the consumer suffers and/or benefits, and the need for systems such as the Fluid Energy Management System are the main thrust of this chapter. The final sections explain how an enhanced version of FEMS would have to operate in a dynamic or multi-tariff environment.

The residential consumer faces some fundamental changes in the way it considers electric energy. In the current climate, electric energy can be seen as a commodity that can be bought, sold, and traded, taking into account its time and space varying values and costs.

The market commodity of electrical energy is based on a 'spot price'. In general terms an hourly or half-hourly spot price (in dollars per kilowatt-hour) reflects the operating and capital costs of generating, transmitting and distributing electrical energy. It varies per half-hourly or hourly time-unit, and in some countries like the U.S. form place to place. In general, the wholesale market place, in this country known as the New Zealand Electricity Marketplace (NZEM), involves a variety of supplier-customer transactions (ranging from half-hourly varying prices to long-term, multiple-year contracts), all of which are based in a consistent manner on the per time-unit spot price.

It is claimed that a spot price based energy marketplace has many benefits for both the electric energy supplier/distributor (also referred to as 'utility') and the end-customers. These benefits include improvements in operating efficiency, reductions in needed capital investments, and customer options on the type (reliability) of electricity to be bought. The intention is that the spot price based energy marketplace is a win-win situation for both the supply/distribution companies and its industrial and residential customers. The customer's lifestyles improve because the customers are receiving more service from the use of electric energy per dollar spent. The utility has a more controllable, less uncertain world in which to operate (Schweppe et al., 1988).

Spot price based energy marketplaces are implemented using today's technologies. However, its existence stimulates the development of new (micro)electronic technologies and hence enables further exploitation of the microprocessor revolution in communication and energy management.

The spot price based energy marketplace concepts were originally developed to meet the present and future needs of the complex, interconnected, sophisticated power systems of developed countries. However, the basic ideas are also applicable to the smaller, rapidly growing, less sophisticated power systems often found in other parts of the world.

The spot price based energy marketplace was developed to be applicable to present-day structures wherein a privately owned utility is regulated by some government agency, or the utility is government owned and operated. However, the energy marketplace introduces the possibility of various degrees of deregulation wherein some generation is provided by privately owned, less regulated companies.

Spot pricing is the natural evolution of existing techniques for power system operation, planning and demand side management.

9.2 Electricity developments in New Zealand

Charging for electricity has evolved from using fixed price contracts to charging by time, until finally, charging by quantity was used following the availability of suitable meters (Westbury, 1994).

In New Zealand approximately 95% of the total electricity requirement is generated and distributed by three state owned enterprises (SOE), previously a single body known as the Electricity Corporation of New Zealand (ECNZ), and Contact Energy. In alphabetical order the four companies are:

Contact Energy Ltd - Separated from ECNZ in 1996 and privatised in 1999. 40% owned by Edison Mission Energy of California (Generation capacity: 2,424 MW; Customers: 355,000).

Genesis Power Ltd - Separated from ECNZ in 1999 and wholly government owned (Generation capacity: 1,594 MW; Customers: 158,000).

Meridian Energy Ltd - Separated from ECNZ in 1999 and wholly government owned (Generation capacity: 2,355 MW; Customers: 72,000).

Mighty River Power Ltd - Separated from ECNZ in 1999 and wholly government owned. Retails under Mercury Energy and First Electric brands (Generation capacity: 1,067 MW, Customers: 271,000).

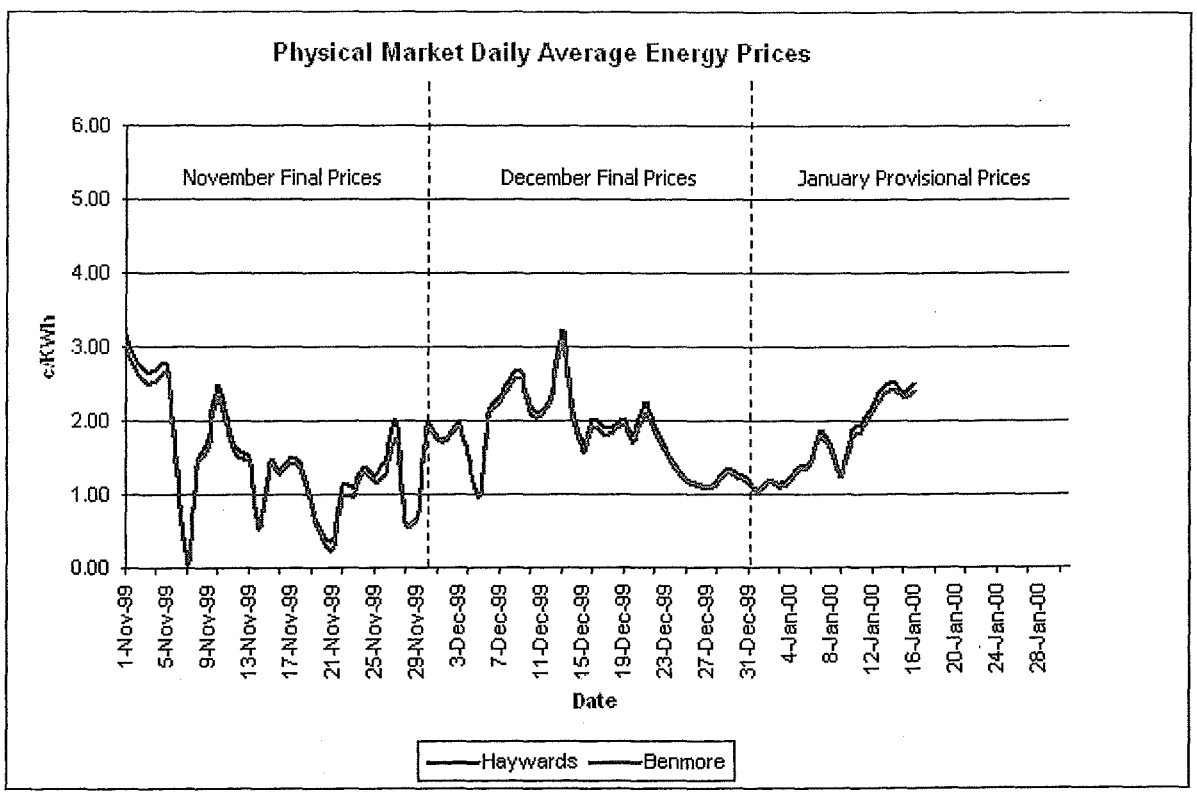


Figure 9.1- **Wholesale energy market prices in New Zealand**

9.2.1 Purchasing electricity

A large proportion of the energy provided by the four companies is purchased by electricity retailers at wholesale prices, which is then distributed and sold to residential, commercial and industrial customers. Retailers and end users have a choice of buying their electricity either off the pool (NZEM) or through bilateral contracts with generators via the Metering And Reconciliation Industry Agreement (MARIA). These two institutional arrangements enable competition in the supply of electricity. An idea of the North Island (Hayward) and South Island (Benmore) market prices and their fluctuation can be obtained from *Figure 9.1.*

9.2.2 Retail tariffs

Retail tariffs used by the electricity retail companies aim to profitably recover the price paid (in half-hour slots) on the electricity market and the costs associated with supplying energy to retail customers, while maintaining a competitive edge within the marketplace.

Due to the limitations of the widely used single register Ferraris disc meter, simple flat rate tariffs, coupled with a fixed supply charge, have been used by retailers for charging residential consumers of electricity. These tariffs do not represent the true cost to the retailer of supplying the energy, and hence have provided no incentive for a consumer to alter their consumption habits. To minimise the bulk supply costs incurred, direct control of consumers' water heating loads has been used to perform *peak clipping.* Customers are rewarded for any inconvenience with a reduced tariff.

As more sophisticated metering and load control equipment has become available, it has become possible for electricity retailers to employ multi-rate tariffs. The most common of these have been in the form of a dual rate pricing option where customers can take advantage of cheaper night rates. Use of this option requires the installation of a dual register meter and direct load control equipment for tariff switching. Use of these tariffs provides a *valley filling* change in load shape (see the section on 'Demand Side Management').

9.2.3 Time-of-use metering

The reforms to the New Zealand electricity industry have introduced competition into the traditional monopoly of electricity generation and supply. This has particular significance to the retail market as it allows large end-consumers to purchase energy from other energy traders or even direct from a generator. Customers selecting this option require half hour time of use metering to be installed at their premises to enable the energy flows to be reconciled between all the parties involved (McGlinchy, 1995; Hunt, 1995; M-co, 1999).

The simplest way to enable retail competition at the *small* consumer end is to operate a *time-of-use meter* in every household. This household will then have the option of either continuing purchasing their electricity from their existing retailer or from another retailer (most likely an out-of-town one). Time-of-use metering would enable the retail supplier to know how much power each consumer uses every half-hour of every day of the year. This would allow special supply and price deals to be put together for consumers encouraging them to use the electricity more efficiently (e.g. by conserving at peak periods). At present consumers without time-of-use meters get charged on a *total* usage rather than a time-of-usage basis.

9.2.4 Profiling

The major problem with time of use metering is that the cost of installing meters for the average consumer is prohibitive at present. The costs would outweigh any benefits (M-co, 1999). Although metering costs are expected to fall, an alternative system is required which is cost-effective in the interim. This is why 'profiling' was developed. Profiling is about estimating a consumer's half-hourly electricity consumption by the use of the 'shape' of that consumer's electricity consumption (their 'profile'). Profiling enables an electricity retailer to calculate the half-hourly electricity consumption of its customers who don't have a time-of-use meter installed. This will form the basis for the retailer to pay for the electricity it purchased from the wholesale market. More importantly, it enables the retailer to sell electricity to consumers anywhere in the country who have not installed the time-of-use meters. Furthermore, under the profiling system, any retailer is able to introduce new profiles targeting specific groups.

For example a retailer may decide to focus on dairy farmers. Meters would be installed by the retailer on a number of representative farms and over time a profile would be built up on how much electricity an average dairy farmer uses at each half-hour of the day. Knowing how much their electricity supply costs are for each time period means the retailer would be able to put together a specific profile just for that group. The same could happen for residential groups such as retirement villages.

Obviously profiling involves using averages and not every consumer will be targeted but it does provide competition to a wider section of the population in a cost-effective manner.

Any system which attempts to profile all 1.7 million electricity consumers in New Zealand is obviously complex. At the centre of the process is the development of a national database, called the Registry, which identifies every electricity meter by way of a unique number or ICP (Installation Control Point). Every retailer has now allocated ICPs to their customers' meters. Since April 1999 these have been printed on the monthly power bill. A customer needs to know their ICP to change electricity retailer (*source*: M-co).

9.3 Demand side management

To understand the usefulness of dynamic tariffs, it is necessary to review the methods that have been employed by electricity supply companies to alter the shape of their load curve, in an effort to control load growth and increase the load factor. Dynamic tariffs can be considered a subset of these methods.

Known as *Demand Side Management* (DSM), the implementation of these methods have given benefits in terms of improved system efficiency and utilisation, and the deferment of the capital expenditure required to increase system capacity (Busch et al., 1996). Additional benefits in the areas of conservation of natural resources and a reduced impact on the environment can also be achieved (Dauncey, 1990).

Figure 9.2 illustrates the six basic options available to an electricity supply company for changing their load shape using DSM, which can be summarised as follows (Talakdur et al., 1987; Hunt, 1995):

- *Peak Clipping* - This involves reducing the load during peak periods and is generally achieved by directly controlling part of a customer's load.
- *Valley Filling* - This method of load shaping involves building load during off-peak periods through the use of special off-peak tariffs.

- *Load Shifting* - Load shifting attempts to move peak loads to off-peak periods without necessarily changing the overall consumption, and represents a combination of peak clipping and valley filling. This form of load shape change allows the most efficient use of capacity and is generally achieved through indirect control using time dependent tariffs.

- *Strategic Conservation* - Strategic conservation involves reducing overall electricity consumption by altering specific patterns of use. This type of load shape change can be achieved through the promotion of energy efficient measures such as the use of building insulation and energy efficient appliances.

- *Strategic Load Growth* - Strategic load growth produces a general increase in consumption beyond that achieved by valley filling. This is achieved by increasing the market share of loads that can be served by other fuels.

- *Flexible Load Shape* - The load shape can be made flexible by offering customers price incentives for reduced levels of service. Examples of a reduced level of service include interruptible supplies and limiting the power and energy that a customer can extricate.

Until recently, most supply authorities (utilities) did not consider DSM options in their system planning process because efficient control technologies were not developed to a state where they were reliable and economic. DSM options including programs to actively reduce load by utililty control, installation of efficient appliances, insulation, lighting, and rate mechanisms (time-of-use) have surfaced primarily in the last 10 to 15 years. DSM introduces additional cost factors that the supply authority need to be aware off, e.g. installation costs, incentive rates, lost energy, rebates, losses savings (demand and energy), transmission and distribution effects, and others (Gustafson et al., 1993).

9.3.1 Load management

Load management, which is a subset of demand side management, is used to produce the load shape changes associated with peak clipping, valley filling and load shifting. Electricity supply companies have mainly implemented load management using *direct* control, which has involved communicating switching signals to consumers' premises. Generally, this method of control has only enabled peak clipping and valley filling load shape changes to be achieved. In recent times, as electricity supply industries have become deregulated (Westbury, 1994), increasing emphasis has been placed on the use of *indirect* load control using *time dependent tariffs*, to achieve the more efficient load shifting shape change. If the time dependent tariffs employed can be varied *dynamically* to reflect the true cost of supplying a unit of energy, efficiency can be optimised (MacDonald et al., 1994; Schweppe et al., 1988).

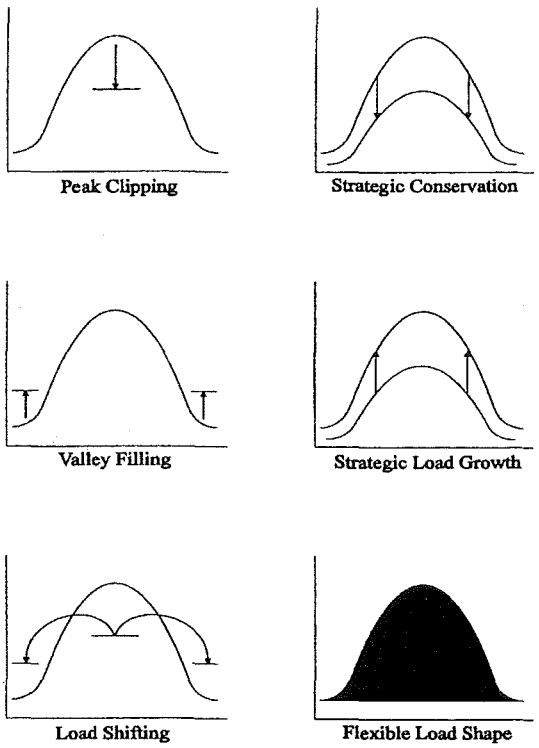


Figure 9.2 **– Six basic options available for changing load shape.**

9.3.2 Direct load management

Direct load management is used to produce *peak clipping* and *valley filling* load shape changes, and involves directly switching part of a customer's load. Most direct load control schemes can also be used for emergency load shedding. Water heating, air conditioning and storage space heating are the most commonly controlled loads. The implementation of direct load management schemes requires a means of broadcasting the switching signals to the customer. As summarised in Hunt (1995), a variety of technologies have been used for this purpose which include:

- *Time-clocks* - These are pre-set with the required switching times, but can become inaccurate and are inflexible. In addition emergency load shedding is not possible.
- *Voltage or frequency control* - This technique involves varying the system frequency or voltage magnitude. Equipment installed on the consumers' premises is used to detect the variations and perform a switching action (Tolley, 1987). This communication method can utilise existing transformer tap changers and frequency control equipment, but is only suitable for simple control schemes.
- *Waveform distortion* - Waveform distortion techniques include cyclo-control (Forrest et al, 1982) and Sequential Waveform Distortion (SWD), (Foord et al., 1990). Cyclo-control involves applying a momentary short, in the region of the zero crossing, to the low voltage side of a supply transformer. The resulting modulation of the supply voltage waveform is detected and decoded by receivers installed in the customers' premises. This technique offers a fast response, a large range of coding options and employs relatively low power injection equipment. SWD involves distortion of the voltage and/or current waveform and is capable of supporting duplex communication.

- *Power Line Carrier* - Power line carrier systems involve the use of radio frequency signals injected into the distribution network using various carrier frequencies and modulation schemes (Nunn et al., 1992).
- *Telephone or other land line* - This technology makes use of separate communication cabling which is generally time-shared with other services to reduce costs (Talakdur, 1987). In the case of telephone lines, the broadcasting capabilities are limited which makes the system unsuitable for emergency load shedding.
- *Radio Control* - This involves the use of radio controlled switches within the customers premises (Fidgett et al.,1987).
- *Ripple Control* - Ripple control involves superimposing narrow-band coded audio frequency pulses onto the mains waveform, which are detected and decoded by receivers installed in the customers premises. Ripple control offers a wide range of coding options and high reliability, but requires relatively high power injection plant and has a relatively low response time.

The voltage/frequency control, waveform distortion, ripple control and power line carrier technologies help to maximise the use of the distribution network by employing it as a communication medium in addition to supply distribution.

9.3.3 Indirect Load Management

The problem with the *time-of-use tariffs* as used in direct load management is that they only represent a forecast of the continuously varying true cost of supplying a unit of electrical energy. The theory of *dynamic tariffs*, or *spot pricing*, has received considerable attention over recent years and it has been shown that an electricity system will only meet *optimum efficiency* if consumers pay the true minute by minute marginal cost (McDonald et al, 1994; Schweppe et al., 1988).

There is some contention as to whether to use short run marginal costs, which vary with temporary changes in demand, or long run marginal costs, which vary with permanent changes in demand. Tromop et al. (1995) presents the case that, for an optimally designed power system, the short run and long run marginal costs are the same. In general terms the marginal supply cost is dependent on factors such as the total demand; consumer load patterns; the operating costs of the most expensive generating plant in operation; the current generation or supply constraints and the current distribution costs (Hunt, 1995).

Considerations that need to be made when implementing dynamic tariff structures, apart from customer response, include the issues of communication and metering. The frequency at which prices are updated will affect both the communication and metering requirements, which will generally become more expensive as the frequency of price-updates is increased.

9.3.4 Tariff Communication

The implementation of dynamic tariff structures will require advanced communication technology that enable the spot prices to be transmitted in real time between the generating companies, retailers and consumers. As discussed in the previous section, the communication technology required will depend on the frequency at which pricing signals are generated. It must also have broadcast capability to ensure that consumers receive the signals in real time.

A number of the communication methods available for direct load management, as discussed in section 9.3.2, can also be used for broadcasting tariff information. In particular the SWD, power line carrier, ripple control and radio control technologies are valid options as they

support robust and flexible coding schemes. The SWD, power line carrier, telephone and radio control technologies also have the advantage that they can be adapted for duplex communication to enable more advanced functions and customer services to be provided, such as remote meter reading; improved fraud detection; remote supply connection; security services; remote reading of gas and water meters; and obtaining load research information unobtrusively.

Utilising power line carrier technology for duplex communication with customers will provide reliable communication at relatively high data rates, while maximising the use of the distribution network (Nunn et al., 1992).

In the interim, tariff communication using ripple control offers a cost effective solution for New Zealand ESA to set up and evaluate indirect load management schemes employing dynamic tariff structures. This can make use of the existing load control equipment while still providing the provision for performing emergency load shedding.

9.3.5 Domestic Energy Management Systems

Complex dynamic tariff structures will help to provide incentives to customers to shift their load from peak to off-peak hours. However, utilisation of these tariff structures requires sophisticated metering and energy management apparatus so that tariff information can be comprehended by the consumer, and subsequently used to optimally control their load. MacDonald et al. (1994) is of the opinion that this control will be implemented most effectively using automation via a distributed control network, as it will enable manufacturer optimised control algorithms to be developed for each individual load. In particular, control algorithms that can anticipate the likely levels and trajectories for the spot prices will provide the best performance.

An important part of any domestic energy management system is the user interface. This must be able to hide the complexities of the tariff structure in use and communicate useful information to the consumer, such as the cost of operating a particular appliance. In addition it must enable the consumer to set preferences governing the operation of their various loads and appliances (Hunt, 1995).

A number of centralised and distributed domestic energy management systems, programs, and models have been developed for evaluating the impact of dynamic tariff structures (Hunt, 1995; Gustafson et al, 1993; Staufer et al, 1991; Dick et al., 1990; Matty, 1989), however further development and evaluation of optimal control algorithms is required.

9.3.6 Alternatives to Demand Side Management

DSM programs are not the only mechanism for realising end-use efficiency improvements. *Appliance and equipment efficiency* standards are having a significant impact on electricity demand in the United States. Standards already adopted are expected to have lowered national electricity use 3% by 2000 (Geller et al, 1999). Some energy efficiency measures, such as power-managed personal computers, "sell themselves" to a large degree. They have been widely adopted without financial incentives or much utility involvement. And energy service companies are increasing the level of efficiency improvement occurring largely through the private sector. While all of these paths to greater energy efficiency are important, there remains a key supporting and complementary role for power utilities to play in promoting cost-effective electricity conservation.

9.4 The energy market place

The introduction of an energy market place can be viewed as the logical evolution of the present day practice in any of the three fields: rate-making, load management, and power system operation. There is always resistance to change; power company personnel are used to making decisions without detailed consideration of rates and customer reactions. Customers who benefit from cross-subsidisation under the normal rate system will be not be keen to pay a fairer share of the costs. Industrial consumers might have to change their plant method of operation to respond effectively to price changes. Basically, the largest obstacle to changing over to an energy marketplace is the (unknown) cost consequence for the consumer.

It is possible however, to implement an energy marketplace by means of a gradual transition from the present system (Ghosh et al., 1997). For example, implementation can start with large industrial consumers and be extended to smaller consumers on a voluntary basis. This allows present and future participants a period of time to become familiar with the new system.

The hardest part of market liberalisation is the reform of the retail supply to small business and domestic consumers. Barton (1999) states that there are two main alternatives for promoting consumer choice: automatic metering and load profiling (see also *Sections* 9.2.3 and 9.2.4). Because of the perceived cost of installing new meters, several countries are turning to determining a load profile for a group of customers for a specific period (Electricity Association, 1998; M-co, 1999). The disadvantages associated with this method appear to be:

- Individual customers do not pay exactly for what they use, they pay pro-rata for the whole group.
- Suppliers will select and nurture those customers with advantageous profiles only.
- It is *not* spot pricing.

Without competitive pressure the supplier has little incentive to improve services or reduce prices. This appears to be the situation in New Zealand even though in theory consumers do have the right to switch suppliers. Telecommunication experience shows that many residential customers will stay loyal to incumbent suppliers even when cheaper alternatives are available (Electricity Association, 1997), because it is hard to understand complicated tariff structures and the risk of switching does not seem worth the effort. Technological developments may allow the market for the small electricity consumer to come into its own. Convergence between internet, telephone and television, on a mass-market scale, is eminently foreseeable. When it occurs, consumers will manage their electricity usage patterns using smart appliances for washers, clothesdryers and water heating, which will only switch on when dynamic tariffs fall below a given level.

The introduction of power system deregulation and market liberalisation in the various countries such as the U.K., Norway, the U.S., Australia and New Zealand has raised the issue of *reliability*. A customer potentially has a wide choice regarding suppliers. Some customers may be prepared to pay more to ensure a higher reliability factor with regards to an uninterrupted supply of electrical energy, and others may be willing to pay less for lower reliability. In this new situation, system planners have to re-examine their load shedding technique and base it not only on past experience but also on customer considerations regarding their reliability costs (Wang et al, 2000).

9.4.1 A wholesale electricity market

In New Zealand the change to a de-regulated energy marketplace began in the electricity sector in 1987, when the government-owned generation and transmission department was corporatised. Key events were:

1987: Government's generation (97% market share) and transmission department was corporatised (as ECNZ Ltd., the Electricity Corporation of New Zealand) and electricity generation deregulated. However, ECNZ's market dominance resulted in negligible new generation by others before 1996.

1992 (Jun.): The 'Energy Companies Act' requires all electricity activities of local government to be corporatised.

1992 (Aug.): A voluntary industry study (WEMS, 1992) recommends to Government the creation of a wholesale electricity market.

1993: Consumers below 50,000kWh per annum become eligible for competitive supply, based on half-hourly metering. Predictably, metering/data management costs ensure that negligible competition emerges.

1994: All consumers now eligible for competitive supply, based on half-hourly metering. Competition proves effective for large consumers and increases to 0.1% of all consumers and 6% of total energy sold.

1995: Government's own study (WEMDG) recommends creation of a wholesale electricity market.

1996 (Feb.): Approximately one third of ECNZ is split off, to become Contact Energy Ltd., and an interim half-hourly wholesale electricity market commences. Government now owns 2 generation companies, with combined market share of approx. 96%.

1996 (Oct.): Final wholesale electricity market commences, and features full nodal pricing.

1997: Retail competition remains at the 6% level achieved in 1994. Few believe that real competition exists at either the wholesale or retail levels. Government promises action "next year".

1998 (Jul.): Government passes the *Electricity Industry Reform Act*, requiring a maximum cross-ownership between 'wires' and 'electricity generation and retail sale' of 10%. In the following 8 months all 34 electricity distributors sell their generation assets and retail businesses; most are bought by the government!

1999 (Apr.): The government-owned Electricity Corporation splits in three, and the earlier-separated Contact Energy is sold. Government ownership of generation reduces to approximately 60%, but its ownership or electricity retailing has increased to a similar market share (source: *Dunedin Electricity*).

M-co Ltd operates the New Zealand wholesale electricity market under contract. It administers the New Zealand Electricity Market (NZEM) and the Metering And Reconciliation Industry Agreement (MARIA). As mentioned previously, these two institutional arrangements enable competition in the supply of electricity. M-co was established in 1995 under joint ownership by ECNZ, Transpower and ESANZ (the now-defunct Association of electricity distributors). In 1999 M-co was purchased by Rand Merchant Bank Australia.

TransPower NZ Ltd is the owner and operator of the national transmission grid. It is a State Owned Corporation, with the Ministers of Finance and of State Owned Enterprises as the shareholders on behalf of the NZ Government.

The Electricity Industry Reform Act 1998 limits cross-ownership between electricity distribution assets and either electricity generation or electricity retailing activities to a maximum of 10%. The term 'Line Business' is used to describe the separated delivery function and there are presently 30 line businesses serving New Zealand's 1.7 million electricity users.

9.4.2 Competitive behaviour

NZEM prices are determined by the competitive action of buyers and sellers in the market. Around 80 – 90% of the total volume of New Zealand's electricity is traded through the market with the remaining 10 – 20% being covered by bilateral contracts between generators and consumers. There are 480 nodal grid exit and injection points in the New Zealand electricity system. Pricing information is collated via the internet-based electronic Commodity Market Information and Trading system (COMIT) on the 244 nodes that are included within the ambit of NZEM.

Prices are calculated for 48 half-hourly trading periods every day of the year. The nodal pricing methodology adopted by NZEM establishes a price for energy, a price for reserves and the magnitude of marginal transmission losses for each node.

In simple terms, the underlying price is set by the intersection of the demand curve (set by purchaser demand) and the supply curve (established by generator offers) for energy. The price varies across nodes according to transmission losses and grid constraints. This enables more effective risk management on the part of Market Participants and also sends strong investment signals to generation and transmission owners.

There are four key price calculations (*Figure 9.3*):

Forecast Prices - calculated from 1pm the previous day up to two hours before the specific trading period. It incorporates:

- energy & reserve bids, and offers entered into the market through COMIT
- a demand forecast compiled from bids.

Dispatch Prices - calculated at, or just prior to, the time electricity is actually dispatched. It incorporates:

- revised energy and reserve offers entered into the market through COMIT
- expected demand.

Provisional Prices - calculated for all the trading periods of the previous day. Published to Market Participants by 9.00 am on the morning following the day's trading and includes provisional demand data. It incorporates:

- final energy and reserve offers
- actual metered demand (provisional).

Final Prices - calculated by the eighth business day of the month for the previous month. Final prices are those used in the settlement of the NZEM. These are available at the end of each month following Grid Operator corrections to metering and grid configuration data.

The history of NZEM shows provisional prices are extremely close to final prices. Accordingly, NZEM is currently considering a proposal to considerably reduce the time delay in publishing the final price to one day only (M-co, 1999).

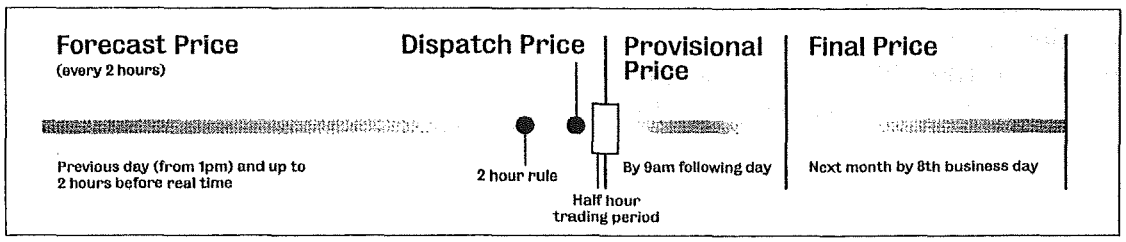


Figure 9.3 – **Timeline for the four key prices used by the NZEM.**

9.4.3 A deregulated market model

The concept of a deregulated market place is not new, see Golub et al. (1983), Bohn et al. (1984) and Henney (1987) to name a few. A model of a de-regulated energy market is shown in *Figure 9.4*. It has three major entities:

- A single 'transmission and distribution company' (T&D) that controls the transmission and distribution system, and acts as a middleman in the energy marketplace. It purchases all the energy demanded by the users and periodically collects all payments by the users and pays the generating companies for the quantities of electrical energy produced.
- An unknown number of independent, private generating companies which sell energy to the T&D company. The quantity sold at a particular moment is the decision of the generating company, and is at the current spot price.
- The users (customers) of electricity who buy as much energy as they want from the T&D company.

An alternative structure to that shown in *Figure 9.4*, which has both advantages and disadvantages, involves a single bulk transmission company and a large number of individual local distribution companies, all of whom are regulated (Chao et al., 1997). This is model is closer to the New Zealand concept of the wholesale electricity market.

In the deregulated model the generators are not centrally dispatched. Instead the Market Co-ordinator (basically the role fulfilled by M-co in New Zealand) sends each generator a spot price and each generator decides to deliver if the spot prices paid for electrical energy exceeds the plant's marginal operating costs. The Market Co-ordinator keeps supply and demand in balance by continuously adjusting the spot price. On nights when demand is minimal, the spot price declines until only generating units with low operating costs remain on-line. As demand increases during the morning, the spot price may rise to the point where owners of electrical storage units, which had purchased and stored power during the night, begin selling back energy.

Figure 9.4 shows two other participants: Information Consultants who forecast future dynamic tariffs (short and long term) and Energy Brokers who arrange side contracts to shift risks. For more comprehensive information the reader is recommended Schweppe et al. (1988).

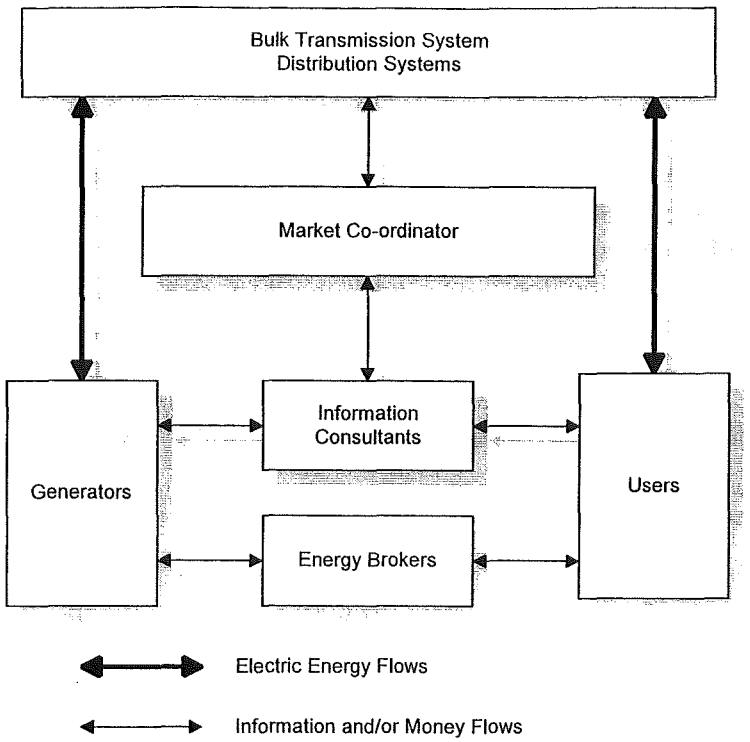


***Figure 9.4* – A model of a deregulated energy marketplace.**

The Market Co-ordinator does not necessarily know the precise operating cost characteristic of the private generators. Hence, the marginal operating cost component of spot prices could be determined empirically by observing how the generation responds to different prices at various times. Alternatively the Market Co-ordinator could ask the private generating firms to provide their operating cost data on a confidential basis to improve system dispatch. As the details of the T&D systems are known, it is then possible to determine the various components that make up a spot tariff from the equations below.

The hourly spot tariff associated with the nth customer during hour t is viewed as the sum of the individual components defined by:

$$P(t) = G_F(t) + G_M(t) + G_{QS}(t) + N_{L,n}(t) + N_{QS,n}(t) + N_{R,n}(t) \qquad (9.1)$$

Where $G_F(t)$ = Generation marginal fuel

$G_M(t)$ = Generation marginal maintenance

$G_{QS}(t)$ = Generation quality of supply

$N_{L,n}(t)$ = Network marginal losses

$N_{QS,n}(t)$= Network quality of supply

$N_{R,n}(t)$ = Network revenue conciliation

'Quality of supply components' arise when generation or network capacity limits are being approached. They serve as curtailment premiums or reliability surcharges. The components of equation (9.1) are often combined into groups such as:

$\lambda(t) = G_F(t) + G_M(t)$ (System Lambda)

$$G(t) = \lambda(t) + G_{QS}(t) \quad \text{(Marginal value of generation)}$$
$$N_n(t) = N_{L,n}(t) + N_{QS,n}(t) \quad \text{(Marginal value of network operation)} \qquad (9.2)$$

The *operating* cost components are usually the largest, being the values for $\lambda(t)$ and $N_{L,n}(t)$. The network components of the hourly spot tariff depends on the customer index n because different customers are located at different parts of the network. This spatial pricing results from the differences in the line losses and the fact that individual lines can become overloaded in one part of the network while over the remaining lines flows are sustainable. In New Zealand, transmission accounts for a relatively large proportion of the cost of delivered energy, especially because the major hydro resources are in the south while the population is concentrated in the north (Read, 1998).

9.5 Consumer response

Why the assumption that both the customer and the power supply and distribution companies will be better off with dynamic tariffs? Theoretically, on the precepts of free market economies the price based on a marketplace sends better signals than do fixed prices. For a rigorous treatment of this theory see Schweppe et al., 1988. Intuitively, the customer can best benefit from spot prices if it has operating flexibility. The higher the day to day and hour to hour variation in price, the more the customer stands to save by shifting its usage. If the customer does not change its demand behaviour, then neither it nor the power utility company are worse off. However, if the customer does respond, the power utility company is better off because all responses will shift energy

According to economic theory, in order to contribute to the collective optimum, a public utility in a monopoly situation must follow the three pricing rules; meeting the demand, minimising its production cost, and selling at marginal cost. This last principle consists of informing the customer on the cost engendered in the supply system by changes in his electricity consumption pattern through the tariff. By selecting the alternative that minimise his cost, the customer will choose the least cost alternative for the community as a whole.

Customers actually have no desire for electric energy as is, but they do desire the *services* provided by electrical energy. Customers respond to different prices in two basic ways:

- ***Modify usage*** – if the price in a given hour is very high, they may reduce usage at that hour because the value of some of the services is less than the price. Alternatively, if the price is very low, they may increase usage to receive services they normally wouldn't buy.
- ***Reschedule usage*** – if the price is high during some hours of the day and though during other hours, customers may reschedule usage so their desires are met but at different times. Such rescheduling usually imposes some customer costs, so the amount of rescheduling depends on the price difference.

If at the individual customer levels, the response to price is a very non-linear, complex phenomenon which depends on the individual customers needs and capabilities. Fortunately, utilities do not need to be able to model, i.e. predict, the response of individual customers to price. The utility is only concerned with *aggregate* customer response. Because of the diversity between customers, relatively simple response models can be used. The classical way to solve for the impact of customer response is to find the intersection of the supply and demand curves, as shown in *Figure 9.5*. If both curves are plotted on the same axis, their intersection yields the resulting values of both price and demand. Both curves change each

hour in response to weather, time of day, outages, purchases from other utilities, etc (Schweppe et al, 1988).

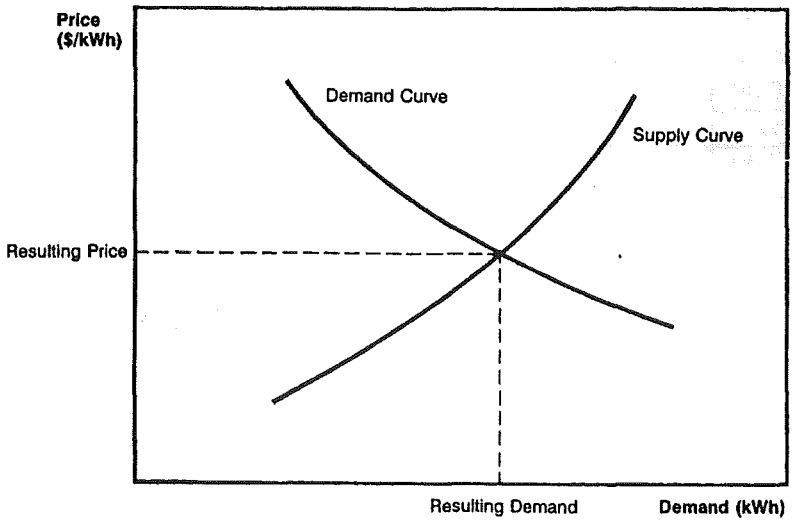


Figure 9.5 **– Intersection of hourly supply and demand curve.**
Supply curve: Marginal cost (price) increases when demand increases
Demand curve: Demand increase when price decreases.

Simulations and trials performed using various dynamic pricing models have demonstrated, by favourable results and consumer reaction, their ability to influence the load shape (McDonald et al, 1994; Kallio, 1992). The overall observation here was that there was little evidence of any variations in the total demand, but the time of day pattern showed some distinct shifts with a move to night-time usage.

In trials in the U.K. involving 500 domestic customers, examination of demand patterns throughout the year demonstrated that customer response to multi-rate tariffs (*time-of-use tariffs*) was most pronounced on winter weekdays, the period when peak demand occurs. These tariffs offered more rates to reflect the daily demand curve, and had different rates dependent upon the time of the year (*Table 9.1*). They are of course not true spot price tariffs because the time tariff relationship is fixed. At the peak (18:30 – 19:00) there was a fall in the usual demand by as much as 40%, with the loads being drawn to late evening and early morning. Nearly 80% of the customers reported in the final survey that they changed their pattern of electricity use. Most importantly from the viewpoint of the domestic consumer was that almost 70% of the customers said that they would like to remain on the tariffs, and 84% felt that savings could be made on the tariffs (Allera et al., 1994)

A qualifying remark here must surely be that the customers volunteering for such pilot programs may be biased in that they may have goals of reducing cost, energy use, or environmental impact; therefore they may not be representative of the general residential population. With this in mind, it becomes difficult to make comparisons with a control group unless it also represents the same subset of the population. There may be considerable uncertainty in projecting the impacts and strategies developed through the trial program to other consumers if the program is introduced on a wider basis.

Weekdays	November and February	December and January	March to October	November to February	March to October
00:30 - 07:30	2.22				
07:30 - 16:00	7.56	9.71	5.24	15.80	4.38
16:00 - 19:00	19.10	40.60	5.24	15.80	4.38
19:00 - 20:00	7.56	9.71	5.24	15.80	4.38
20:00 - 00:30	3.71			4.33	
Weekends					
00:30 - 07:30	2.22				
07:30 - 00:30	3.71			4.33	

Previous tariffs	Standard	Economy
00:30 - 07:30	6.16	2.22
07:30 - 00:30	6.16	6.49

prices in pence/kWh

Table 9.1 – **Time-of-use tariffs as utilised in the U.K. trials (London Electricity, Midlands Electricity, South Wales Electricity, Yorkshire Electricity).**

The French experience, although not of a recent date but still valid as it spans many years, (Lescoeus, 1986) has shown that incentive tariffs can modify consumer behaviour over a period of years to dramatically smooth the load curve. *Figure 9.6* shows that the daily load curve has flattened out considerably. However, when analysing this example one must bear in mind the differences between how electricity is used in different countries. In France a large percentage of the load is industrial, whereas in other countries the majority of the load is domestic and commercial. For a large industrial or commercial consumer the effort and cost expended on monitoring tariffs and on switching loads is small compared to the savings in energy costs. The domestic consumer will only respond to spot prices if the metering and domestic energy management equipment makes it easy to do so. These are the reasons why spot tariff schemes have thus far only been fully utilised with large industrial consumers.

The Germans have a different approach again which is hailed as being more customer friendly (Meier, 1993), although there is no record of how the consumers responded to the changes. In 1992 the German utilities under the auspices of the government introduced a new tariff for the domestic consumer. The tariff now consists of:

- Energy rate (the amount of electrical energy used)
- Demand rate
- Metering and billing charges

New is the demand rate. With a different type of meter the demand over a 96-hour period is integrated and the *highest* consumption is recorded, and is updated hourly. The demand rate encourages the consumer to shift a proportion of their consumption to the off-peak periods during the night. In fact, during the low load period the demand is not metered, so that the customer pays no demand charge for the electricity supplied during that period. In addition the customer will pay a lesser energy rate in that interval, thus winning twice over.

Although there are definite advantages to be gained from adopting spot pricing there are also some negative aspects for the consumers. Amongst these two considerations stand out:

- The difficulty that consumers will have in tracking and responding to a varying price signal.

- The increased metering, management and communications cost that will inevitably have to be born by the consumer.

With respect to the former consideration it is clear why, amongst other reasons, the design of FEMS has opted for the automated black box approach. As for the second reason, it is intended that the consumer can recover these costs from having used less electrical energy.

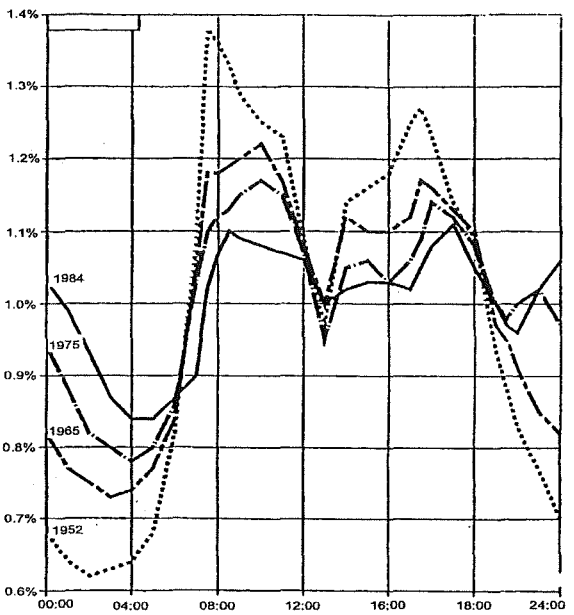


Figure 9.6 – **Change in the profile of the French daily load curve.**

Schweppe et al. (1996) studies and ponders residential response to spot pricing, and comes to the conclusion that electronic support is the key issue. In fact, in the energy consumer market as a whole there is a widely held view that it is not realistic to expect the residential consumer to interact effectively with a spot price based energy marketplace without a well-designed micro-processor based information and control system with a user-friendly interface. Therefore a key issue in residential response is how long it will take before the costs of such electronics becomes less than the benefits (to the consumer and utility) of the resulting response. Some might argue that such a time has already arrived.

9.6 Spot prices

Load management in an electrical power system is achieved through a combination of incentive tariffs and direct load control. Spot pricing is a means of improving the incentive tariff structure by allowing the tariffs to be changed more often. The term 'spot price' means a dynamic tariff that varies as a direct consequence of system load.

The motivation for implementing load management in a power system is the way that demand for electricity is linked to the cost of production. The major costs associated with the production of electricity are the capital and maintenance costs of the generating and distribution facilities. For a given system size, these costs are approximately constant and are not strongly related to the total number of kilowatt-hours used during a year. This is especially true in a hydro-electric system where the fuel, water, is free within the limitations of the total volume available. The power system size, and hence cost, is determined by the *peak* kilowatt load during the course of a year because the system is designed to meet this

load. Conflicting with this fact is a present tariff structure to the end-consumer, which charges them a fixed rate multiplied by the number of kilowatt-hours used. The charging rate per kilowatt-hour used is designed to recover the power system costs on an annual basis. It is therefore evident that by raising the average demand while not increasing the peak, the charge required per kilowatt hour is reduced. Conversely in a single tariff system an increased peak load may not lead to the sale of more kilowatt-hours of electricity over a year.

The benefits to the power system of keeping peak demand down are felt by both the generating and distribution companies. The generating company does not need to build further facilities to meet the peaks, and the distribution company does not need to put in more plant to distribute peak load.

The smoothing of the load curve may be achieved through i) incentive tariffs and ii) direct load shedding. The advantage of spot prices is that they allow for indirect control of more than just domestic water heating and night-storage heater loads. Instead control is achieved over the whole load, as well as allowing consumers to decide what price they are prepared to pay for peak load power. Dynamic tariffs also allow for the filling, by incentive, of deep load troughs that occur during the night. *Figure 9.6* showed how incentive tariffs have managed to smooth the French daily load curve over a period of 30 years.

With suitable technology available the implementation of spot prices for the domestic consumer is greatly dependent upon the expected benefits outweighing the cost, unless the issue is forced. The benefits from any introduction of dynamic tariffs will be of a long-term nature. For the power companies it is more direct, as the need for new capital works is slowed.

The technical problems associated with spot tariffs are in the means of communicating the spot tariff to the consumer and the means of metering how much power was used at which tariff and how customer equipment will respond to tariff changes.

To enable spot pricing of energy units, pricing information on the cost of energy on a per unit basis must be able to be transmitted to a large body of individual consumers and updated on a regular basis. The cost of energy units will depend on the total demand for energy and will vary throughout the day.

For consumers in a power distribution system, the total demand for energy use monitored at a central location or at a number of sub-stations. Based on the total energy consumption measured at a given time, the spot price for energy on a per unit basis is determined. Spot pricing information for the cost of energy as well as timing signals for the appropriate period for which the transmitted spot price is to apply for energy consumed, has to be transmitted to consumers so that the amount of energy consumed can be measured and priced accordingly. Because of the large body of consumers that must receive spot pricing information simultaneously a means of achieving mass communication is required.

At present, in New Zealand, electrical supply authorities can control loads to a ripple control system. At peak load periods the supply authority switches of some of the loads for a certain period. In the ripple control system a 300 Hz signal is superimposed on the mains power lines voltage. The ripple signal will switch devices connected to the consumers ripple control switch on or off. Hot water cylinders and night-store heaters utilise this system. The limitations are that the consumer does not have complete control over their energy use and they don't have enough information to make intelligent decisions. If the spot price of electricity is communicated to the consumer's premises, than consumers can modify their consumption as they deem appropriate. Because the per unit price of energy is varying, the metering system will need to combine of the instantaneous unit price and the present energy

consumption (in kWatt/hour) to work out the total cost of energy that has been used. A metering system with this capability is designated as an *automatic meter*.

Developments in technology have reduced the costs associated with spot tariff metering, but it is still handicapped by an extra cost over conventional metering. This cost needs to be minimised further to attract the consumers. In addition, developments are needed in domestic energy managers for intelligent control over not only the cylinder water heater and night storage heater, but also all the other electrical implements that consume significant amounts of electrical energy. The ability of consumers to easily control their entire load in response to spot prices will make it cost effective to move away from just an incentive night rate to dynamic rates, i.e. the use of spot pricing would create a need to market this principle to the public. This would generate an awareness of the savings that can be made by using domestic energy controllers, without which spot tariffs lose most of their effectiveness.

During a transition phase from fixed to dynamic tariffs the power authority ripple control of domestic water heating would have to continue. After a period of time it may be found that the need to shed water heating loads occurs rarely, because the influence of intelligent energy management systems have flattened the load curve. As a safety measure it would be wise to reserve the right to shed load should the need arise.

There are several different possible types of tariffs, which vary on the basis of the number of tariff rates and when these rates apply.

Time-of-use tariffs - these are not true spot price tariffs because the time tariff relationship is fixed. They are already in existence to a limited extent with the day and night rates offered by the supply authorities. An extension of this method would be to offer more rates to reflect the daily demand curve, and possibly to have rates dependent upon the time of the year. The great advantage of time-of-use tariffs is that they are relatively easy to implement in existing power system and customer metering. It is considered a good first step towards the eventual implementation of fully dynamic tariffs. The disadvantage of time-of-use tariffs is that they do not truly reflect the instantaneous demand for electricity and therefore cannot smooth the daily load curve as well and true spot pricing.

Spot price tariffs - these fluctuate in a dynamic manner as the load varies during the day. This type of tariff will be very effectual in smoothing the daily load curve and could be updated as often as every thirty minutes. It is likely that any fluctuations in the price of electricity during a day will be between pre-set limits to allay consumer's concerns about being charged exorbitant amounts.

This type of spot tariff is reported to have been used by the former Auckland Electric Power Board with some of its large industrial consumers (Andre, 1988).

The advantage of the true spot tariff is that it would be most effective in smoothing the daily load curve and hence keeping the average price of electricity to a minimum. The disadvantage of true spot tariffs is that it necessitates the development of a reliable and nation-wide means of a communicating price to the customer and for the consumer's meter to respond to the price and store the amount of power used at each rate.

9.6.1 Communication for spot tariffs

Time-of-use tariffs and true spot tariffs both require some means of communicating the price information to the consumer. Time-of-use tariffs may be pre-set within a consumer's power meter and then altered when a meter reader visits or could be centrally programmed from the power authority computer. True spot tariffs must have centrally generated information. Several communication methods are available and have been discussed in *Section* 9.3.4.

9.7 eFEMS: forecasting multiple values

It has been made clear from the previous sections that any type of energy management system or controller, in a *real* energy market, is going to have to deal with a significant number of dynamic tariff values during any day of the year. With half-hourly transmitted spot prices this becomes as many as 48 values per day. A system like FEMS will have to incorporate these values in it's prediction and decision making routines, i.e. it will have to forecast a number of hot water draw-offs throughout the day (a demand *profile*) and then decide at which interval(s) during a 24-hour period it heats the water, so as to be able to meet both the hot water demand as well as utilise the low tariffs periods. This new system is termed *eFEMS*, for enhanced Fluid Energy Management System.

The reader should be aware that with this scenario there is no need to predict the *time of first draw-off* (i.e. the case for FEMS) as the various times of hot water use will be patterned in the profile.

9.7.1 Real Time Pricing

There are two categories of dynamic tariff that will be examined with regards to their affect on FEMS prediction. These so-called 'Real Time Pricing' (RTP) categories are:

Day-ahead dynamic tariff – a mild form of RTP where the electricity supplier lets the domestic consumer know in advance what the half-hourly, hourly, or maybe three-hourly spot prices will be 24 hours later. These prices are binding.

On-line dynamic tariff – an extreme form of RTP where the spot price is declared on-line at the commencement of each time-step. The supplier does not make any price declarations binding for several future time-steps.

It is felt that the first category is a possible intermediate step that an electricity retailer is likely to implement before advancing to the last category. The last category presents a final step to achieving true spot pricing. It can only be made if the intermediate step of day-ahead pricing has met with acceptance in the market place. It is felt by the author that, simultaneous with this acceptance, further development must have taken place on intelligent energy management systems (of any sort) so as to get the most out of on-line dynamic tariffs. This last stage can only occur if proper forecasting systems are in place and the energy market has made enough initial data available for the systems to make useful predictions.

9.7.2 Profiles with multiple prediction

FEMS, or more correctly eFEMS, must now work with a profile of spot prices. In the case of *day-ahead dynamic tariffs* this profile should be made available to the system by the supply authority. The profile will consist of up to 48 tariffs and has the following attributes:

- The tariff values are transmitted to the consumer a period of 24 hours preceding the time of applicability.
- The tariff values cannot be altered for the given 24-hour period.
- The tariff values will be stored either in the meter or in eFEMS.

A profile of hot water demand will now have to be forecast by eFEMS for the corresponding 24-hour period. So instead of the single value for hot water demand that has been made available to date, the system must now come up with 48, 24 or whatever the time interval is, contiguous predictions. The actual number of predictions is not relevant at this point; what is important is to establish the means of achieving these multiple forecasts.

There is a general consensus in the neural network community (see *Chapter* 5) that allowing a single neural network to make more than one prediction, although possible, is not recommended for the simple reason that the network will not be able to optimise its structure and weights for a single task. Far better is to have *additional* networks that each concentrate on their area of prediction.

There is also a subtler problem with multiple prediction. The further out into the future the prediction is made, the more error is to be expected. For example, if a prediction is needed for 1 sample ahead and 10 samples ahead then in most cases of practical interest the average error in the 10-ahead prediction will be significantly greater than that in the 1-ahead prediction. Most training algorithms choose networks connection weights in such a way that the average error is minimised. Since the 10-ahead error will be the dominant component in the average error, that component will be favoured in the training process. Weight changes that could produce a *relatively* large improvement in the 1-ahead error will not be made if they increase the *absolute* error in the 10-ahead prediction by an amount exceeding the absolute improvement in the 1-ahead, even though that may represent a relatively small worsening in the 10-ahead prediction. The net result is that the 1-ahead prediction will almost always be inferior to what it could be if it had been made by an independent neural network (Masters, 1993).

That this practical method works in real industrial applications is shown as recently as Feb. 2000 by Russel et al., whom made long-range predictions (20 or more time-steps into the future) for a plant evaporator using sub-neural networks. Each network was used to predict a parameter, which were subsequently combined to form a full predictive model. Not surprisingly, RNNs were used and trained with BPTT, with the Levenberg-Marquardt optimisation algorithm replacing steepest descent. It must be said that although this is not quite the same as *multiple prediction* of a discrete time series, the principle demonstrated is similar.

9.7.3 Prediction with day-ahead dynamic tariffs

For eFEMS this means that a considerable number of separate Elman recurrent neural networks will be needed, one network for each time interval that needs to have a value predicted. If half-hourly day-ahead tariffs are provided and it is decided that this is the time-interval to be used, than 48 neural networks will take up the task of forecasting a hot water demand profile for the coming 24-hour period.

Each recurrent network will have as input the historic values of hot water demand at exactly the same time interval as the one being predicted, i.e. the hot water demand at 2pm on Saturday, Friday, Thursday, and so on. Whether each input or training vector will need 14 or less historic data values plus any Boolean information with regards to day, month, etc. is a matter for trial outcome; much as for FEMS in *Chapter* 8. For each prediction to be made there will also need to be memory set aside for the data storage, i.e. for 56 (historic days) x 48 (half-hour intervals) complete training vectors; although here savings could be made by not duplicating the Boolean information for same day historic data. Unfortunately the net result is still a considerable amount of data storage and processing.

What is needed is some rationalisation for reducing the number of networks required; thus easing the pressure that is being exerted on processing capability and available memory (and hence cost). One solution that comes to mind takes into account the heating time involved for the amount of hot water in the cylinder. A good length of time in which to heat a significant quantity of water is 3 hours. If a demand profile is build up based on 3-hour intervals than this reduces the number of neural networks by a factor of 6. A total of eight recurrent neural

networks would then be sufficient to establish the estimated draw-off over the 24-hour period (*Figure 9.7*).

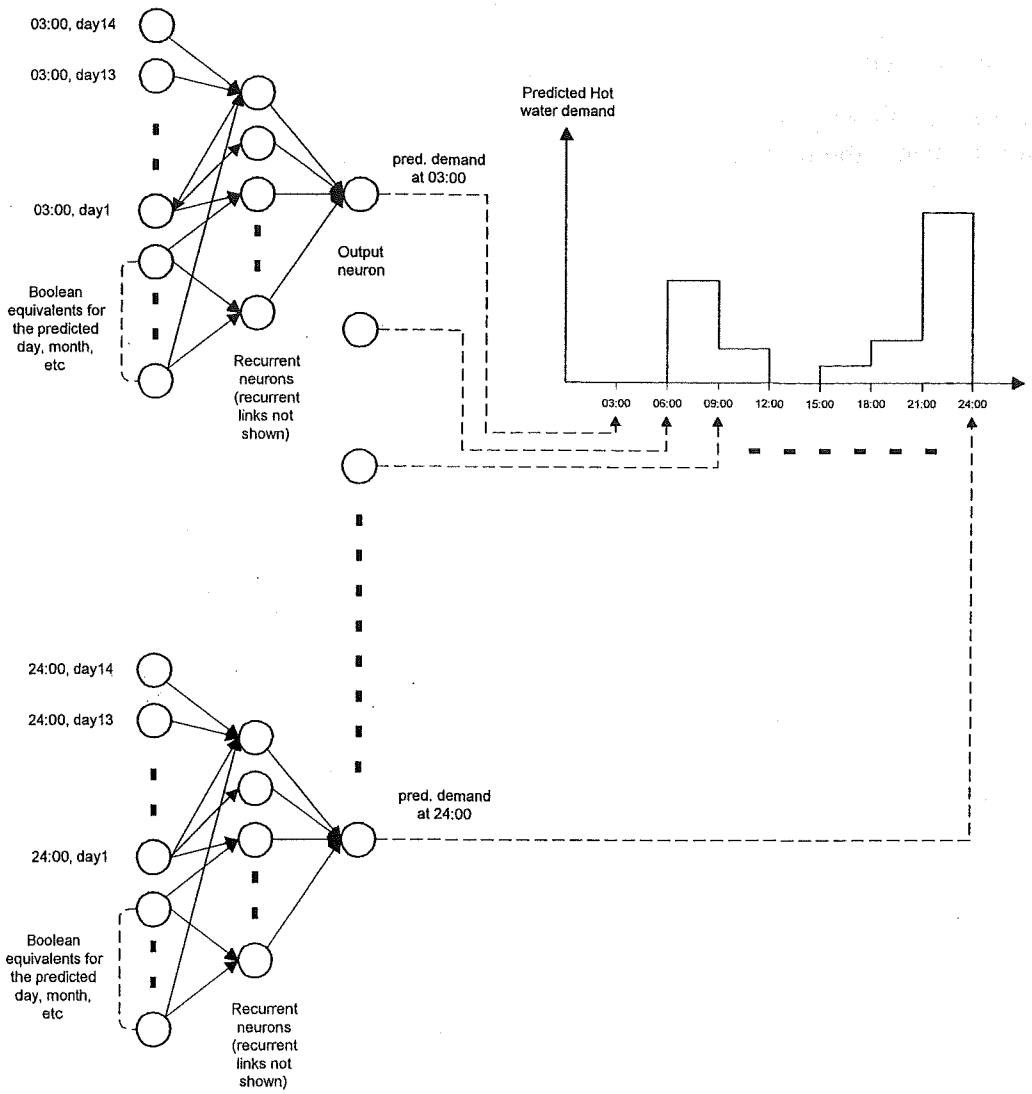


Figure 9.7 **– Multiple prediction of hot water demand using 8 Elman recurrent neural networks for establishing a 24-hour profile.**

9.7.4 Prediction with on-line dynamic tariffs

The situation becomes considerably more complicated when the electricity retailer opts for on-line dynamic tariffs. Now a 24-hour profile needs to be forecast not only for hot water demand, but the dynamic tariffs as well. What will happen in the simplest case is that the demand profile and the tariff profile are predicted, the eFEMS decides when and how much to heat by profile comparison, and will stick to this heating pattern regardless of the dynamic tariff price behaviour. But this is the simplest case, and other options are possible; for instance one where the customer specifies beforehand that cost-saving takes precedence over hot water availability (see also the following section). In such a case the eFEMS will react to the on-line tariff and postpone its heating interval if an erroneous forecast had been made. In actuality a mismatch in prediction and actual tariff is unlikely to be common, as the electricity retail industry will try and avoid wildly fluctuating dynamic tariffs for fear of creating a bad impression on its customers; one of lack of control. A schematic of how

eFEMS would utilise on-line dynamic tariffs and hot water demand is shown in *Figure 9.8* where the tariff data is made available to eFEMS via an unspecified decoder, which converts the tariff level signal sent by the regional distribution authority to values understood by the system (Wezenberg et al., 1995)

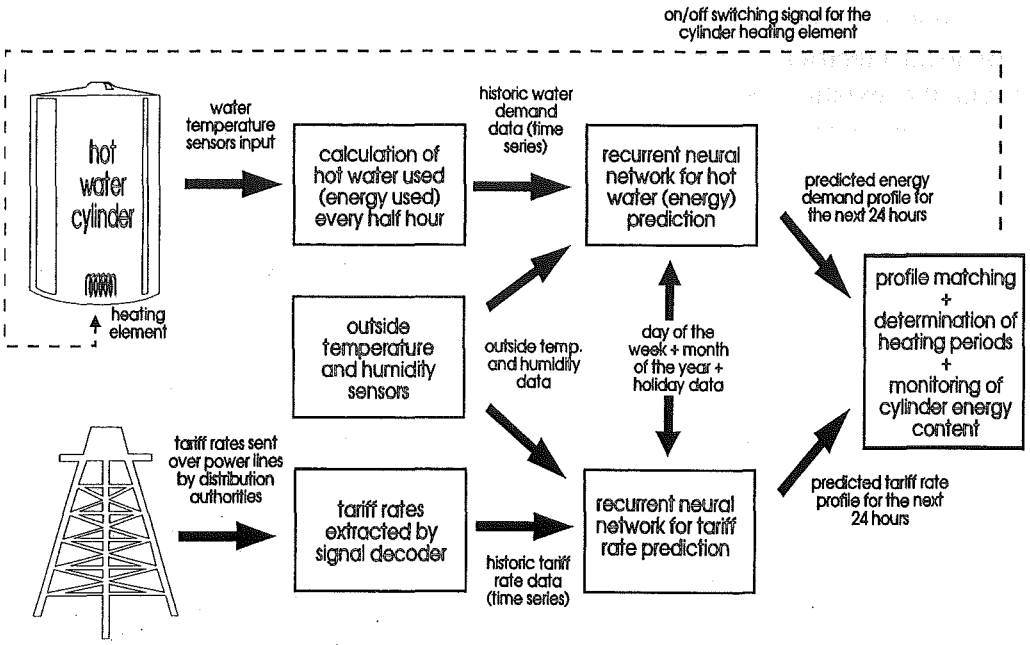


***Figure 9.8* – eFEMS: the role of on-line dynamic tariffs and the various predictions in heating the cylinder.**

9.7.5 Profile modification

The need for tariff forecasting is apparent when it is remembered that large quantities of water cannot be heated instantaneously; the system thus cannot afford to wait for the on-line tariff to go low, start heating, only to have to switch off half an hour later because the newest tariff exceeds the limit the customer is prepared to pay. It also would have no idea as to when the tariff will go low again. Heating up the quantity of water predicted to have it available at the predicted moment would become a matter of chance.

There is also the question of whether or not to modify the profiles as actual usage and tariff information becomes available during a 24-hour period. With regards to tariffs the answer can be affirmative, as this was already briefly discussed above where degrees of cost saving versus satisfying the hot water demand came into question, and is given more detail in the next section on the usage of eFEMS.

As far as modifying the hot water profile, this is a moot point. Constantly modifying the profile when draw-off does not occur at the predicted time could lead to the algorithm deciding not to heat the original forecasted quantities of water. This is a dangerous path to take. If a forecasted amount of hot water is to be used by a certain time, say 6am for a shower, and this has not happened, than it cannot be simply dismissed as never taking place. It is well possible that the withdrawal could occur later towards the evening. The predicted profile for water demand is therefore best left alone.

9.8 Using eFEMS with dynamic tariffs

As discussed in the previous section, an enhanced version of the FEMS should be ready for the time when the electricity supply scenario will include not just a night tariff rate but a number of dynamically varying tariff levels. These rates will reflect to a certain extent the price the distribution authorities have to pay for power on the spot price market as well as the influence of local electricity demand. It is anticipated that rates will be transmitted to the consumer via the existing power distribution network by means of any of the methods outlined in the section on Demand-Side Management, i.e. ripple control, SWD, etc.

Figure 9.9 shows three stages that a local supply authority could follow in order to move from a day/night rate situation to a real-time dynamic tariff environment. The first stage uses *time-of-day* tariffs with seasonal variation. Only one of the envisaged 4 or 5 price levels would be applicable during a particular time interval in 24 hours. This is much like the trials undertaken in the U.K. and elsewhere (see the section on Consumer response). The intermediate stage, *day-ahead* dynamic tariffs, is a logical development of the earlier stage. Here the assumption is that newly developed metering and management systems are already able to receive the 4 or 5 or more price levels from stage one, and will experience no problems if the frequency of price updating occurs not a seasonal but a daily basis. Consumer confidence and metering/ management equipment sophistication will need to have reached a comfortable level before making the move to the last stage, *on-line* dynamic tariffs, thus completing the shift from a fixed energy pricing system to variable market dominated pricing system.

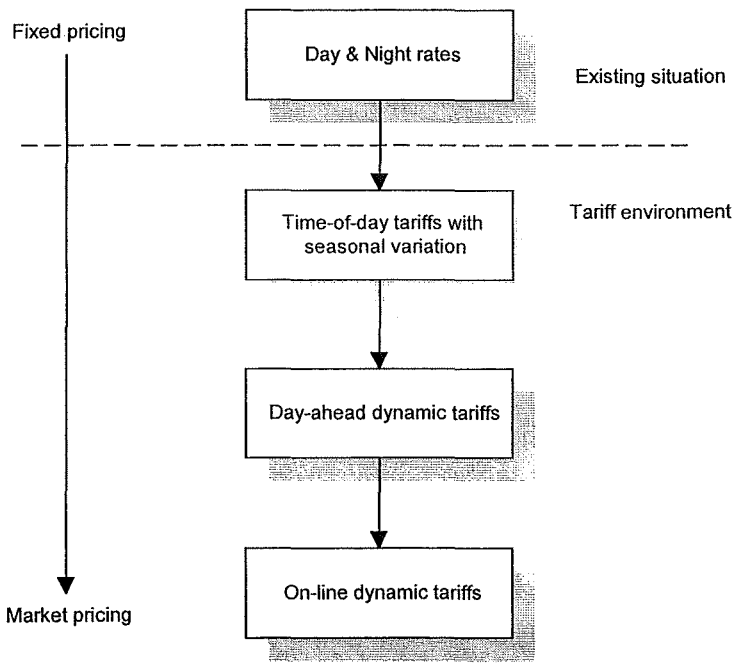


***Figure 9.9* – Introducing dynamic tariffs to the domestic consumer: development stages**

The actual rates, the number of tariff levels, how often they'll change in a 24-hour period, and how far ahead of time they'll be made known to consumers has not been resolved on a local level in New Zealand, and might well vary from one electricity supplier to another.

9.8.1 Cost/demand commitment options

It is envisaged that the consumer will indicate to eFEMS the financial commitment he is prepared to make. With this in mind the system gives the consumer a choice of three cost/demand options, ranging from 'Economy dominant', and 'Balanced cost/demand' to 'HW demand dominant'.

An *Economy dominant* system will concentrate on delivering the forecast hot water demand by only using those time slots that have the cheapest tariff level e.g.. levels 1 and 2 in *Figure 9.10*. The risk here is that the user could run out of hot water. With a *HW demand dominant* choice the system will still work with the cheap tariffs where convenient but will concentrate on meeting the hot water demand at all costs, even if heating needs to take place in the more expensive time slots. The Balanced cost/demand option allows the system to use all the tariff levels from 1 to 3 for heating, ensuring that hot water is available for the normal pattern of usage, but risking shortage if irregular high demands are placed on the system.

The net result of this approach is that with the first case - economy dominant - the consumer will have a low heating bill but faces the possibility of not having enough hot water to meet the needs; whereas for the latter case - HW demand dominant - the hot water quantity will be there, but so will a possibly higher heating bill.

9.8.2 Matching profiles

How eFEMS might process and match the dynamic tariffs and the hot water demand profiles to its advantage is explained with an example using day-ahead dynamic tariffs.

Based on real load information from a local supply authority in New Zealand, a typical load profile for a cold winter's day with daytime temperatures averaging around 5°C is shown in *Figure 9.10* for a 24-hour period. The profile has been split up into three hourly periods, and each period has been relegated an assumed specific tariff. There are a total of five tariff levels.

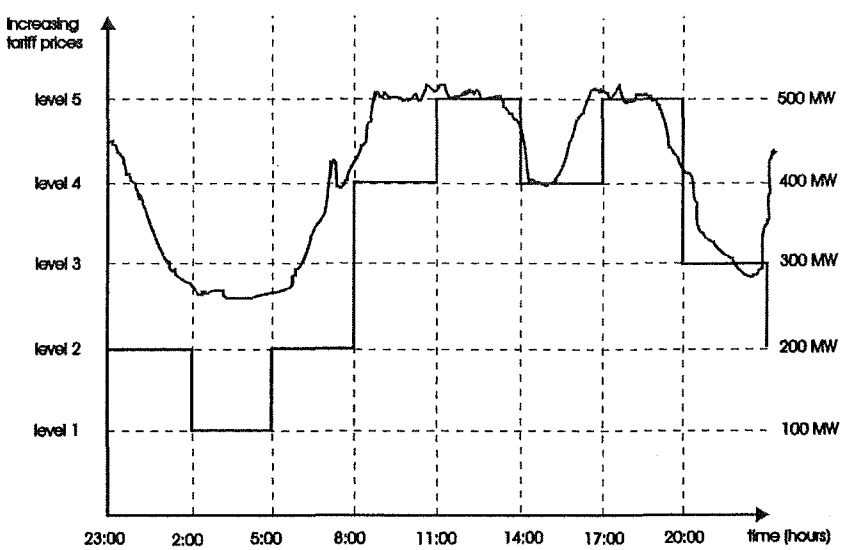


***Figure 9.10* – An actual power demand curve during a winter day (temp. < 5°C) and suggested tariff levels.**

As the supplier forecasts the load curve for the next 24 hours, it bases the day-ahead dynamic tariffs on the load forecast and possibly modifies the tariffs to reflect other factors e.g. expected power outages, energy reserves, etc.

If the consumer has chosen for the option 'Economy dominant', then eFEMS, having obtained both the tariffs and hot water demand for the 24-hour interval, must determine the time slots that coincide with the lowest tariffs of level 1 and level 2. Observing the profiles in *Figure 9.11* shows that this is a single 9-hour interval from 23:00 on the previous day stretching to 08:00 on the day itself. First draw-off takes place sometime between 05:00 and 08:00, and as there are no other intervals where the tariff drops down to a usable level, eFEMS will be forced to heat the *total* volume of the predicted hot water demand *before* 05:00. This despite the fact that the largest demand obviously takes place at a much later time, after 20:00. (The HW demand profile, by the way, is an actual pattern from a day in June, and the tariff profile is the one assumed in *Figure 9.10*). The situation would not have changed even if the consumer had relaxed his preference for economy to any of the other two cost/demand options.

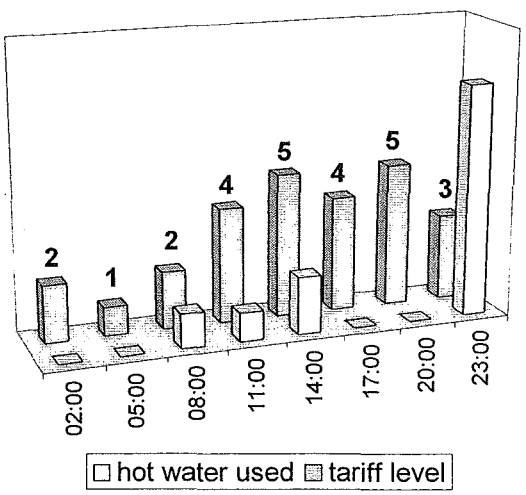


***Figure 9.11* – An example of a hot water demand profile and a *winter* dynamic tariff profile for a 24-hour period, split into 3 -hour intervals.**

But another example with a less extreme power demand profile offers better opportunities. *Figure 9.12* shows a load profile for a summer's day with the same distribution company. The peak demand is significantly lower and with the simple tariff level representation that has been assumed there is a corresponding decrease in the tariffs. Using the same hot water demand profile as in the previous example it is possible to arrive at a more efficient heating process.

With 'Economy dominant' as the selected eFEMS option the volume of the draw-off between the period 05:00 and 14:00 can be heated using the level 1 tariffs prior to 08:00 (*Figure 9.13*). The quantity of hot water that is used *after* 20:00 can be heated with the level 2 tariff preceding and during that same period, thus avoiding standing losses. Of course for both the examples no actual cost has been linked to the tariffs. If in the last example the level 1 tariff was substantially cheaper c.f. level 2, than it might more cost effective to heat the 20:00 evening draw-off (plus an extra to account for standing losses obtained until 20:00) with the lowest tariff available between 05:00 and 08:00.

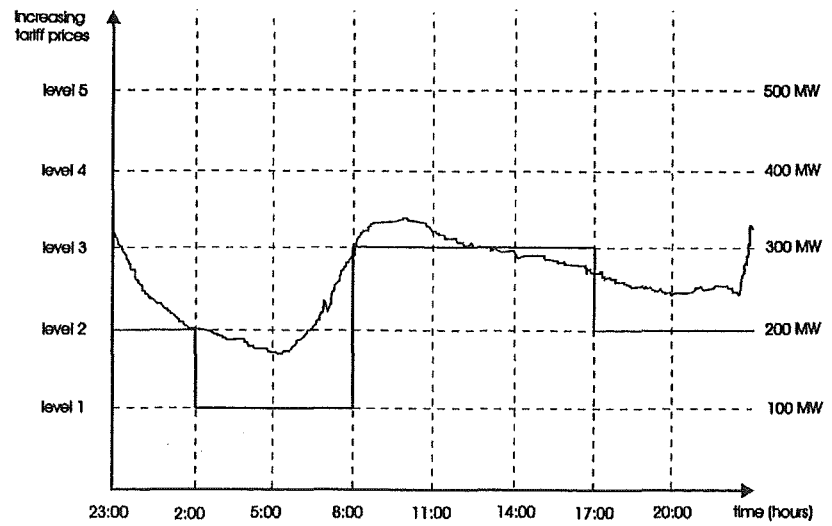


Figure 9.12 - **An actual power demand curve during a summer day (temp. > 15°C) and suggested tariff levels.**

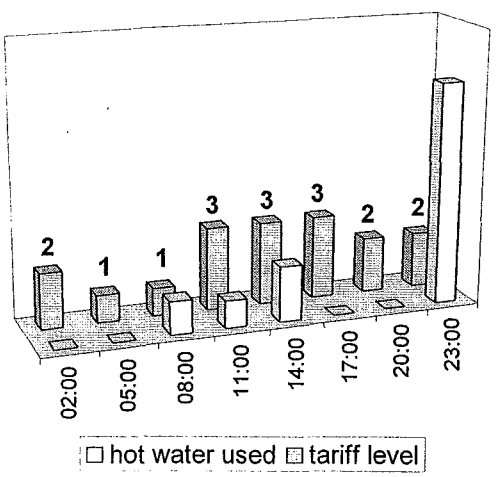


Figure 9.13 – **An example of a hot water demand profile and a *summer* dynamic tariff profile for a 24-hour period, split into 3 -hour intervals.**

Although day-ahead dynamic tariff has been used for demonstration, the principle remains the same for on-line tariffs. The only difference being that the tariff levels would have been forecast instead of transmitted.

9.9 Summary

Deregulation of the electrical generation and distribution industry in an increasing number of countries will lead to substantial cost savings for all classes of consumers, and a plethora of new products that will take advantage of the dynamic retail tariffs to be introduced.

Presented is an enhanced Fluid Energy Management System (eFEMS) which utilises a number of Elman recurrent neural networks for forecasting local tariff rates and domestic hot water energy demand. When both future tariff rates and future hot water energy demand are known the enhanced system determines the optimum heating periods and switches the element accordingly. The system aims to level out daily load demand for the electric power generating and distribution companies and to serve the consumer by cost-effectively heating a

bulk hot water supply. The system is equally well applied to large-scale industrial situations with similar modes of heating.

Chapter 10. Conclusion

This thesis has looked at the potential of improving the control and providing a more cost-efficient operation in the direct heating of stored domestic or industrial fluid mediums; having studied and established the inherent possibilities, an intelligent automated energy management system was deemed a credible solution and subsequently designed, built and tested.

Research started by reviewing the literature on hot water storage and control, specifically in the area of the hot water cylinder as found in most New Zealand domestic dwellings. It was assessed that, besides domestic households, commercial and industrial facilities with substantial quantities and varieties of fluid mediums offer abundant scope for applicability. It appears both areas can obtain significant financial savings with improved energy management; and both consumers and power supply/retail authorities will benefit with increased utilisation of cheaper 'off-peak' electricity; reducing costs and spreading the system load demand.

The domestic hot water cylinder's basic function is to store water at an elevated, predetermined temperature, so that a volume of water can be drawn off for immediate use. The hot water used is then replaced in the cylinder by cold water and a heating element, controlled by a simple thermostat, maintains the water at the pre-set temperature.

It turns out that this system has a number of disadvantages. Once fresh cold water is introduced into the container, the existing hot water layer deteriorates and intermixing occurs, essentially decreasing the available volume of water at a high temperature.

Another difficulty with the cylinder is that the thermostat and the heating element reheat the cold water as soon as the thermostat registers a temperature drop. It takes no account of the time of the day. Thus the thermostat may be causing the heating element to draw power at a time of day when the power is most expensive and in all probability before the heated water is actually needed. Also, known thermostats take no account of the hot water requirements or hot water use habits, and always heat a set volume of water regardless.

It was established that domestic energy control methodology and equipment has not significantly altered for decades. Given the versatility and relative lack of cost of the current microprocessors based hardware and accompanying software, the way was paved for the application of computer technology in this area of great potential. Today's technology allowed the design and implementation of a 'Fluid Energy Management System', simply referred to as FEMS, which took advantage of the presently established day- and night-rate tariff system as can be found in most of New Zealand.

As a first step the research gathered data on fluid behaviour in medium- and high-pressure domestic hot water cylinders; an area not well covered to date and of interest to engineers and manufacturers alike. For this step data acquisition equipment and software was purposely created. The control software plus equipment were combined into a fully automated test system with minimal operator input, allowing a large amount of data to be gathered over a period measured in months.

Typically the effect of a number of internal and external parameters on cylinder thermal behaviour were determined i.e. the influence of different draw-off rates and volumes, standing losses, water mixing, element heating rate, interrupted heating, and element position.

A paper has been written on this aspect of the project and is to be submitted to the IEEE Power Eng. Society for possible transaction publication.

With the gathered information and based on the automated test system, data acquisition software was developed which replaced the existing cylinder thermostat, took care of the necessary data-acquisition, controlled the cylinder's total energy input and logged the losses. Sections of this data-collecting program formed the basis for the later FEMS software; this software providing greater accuracy and safety, a degree of flexibility, improved feedback, legionella inhibition, diagnostic capabilities and potential for future utilisation of the dynamic tariffs, the latter to be offered in a retail energy market for small and medium customers.

A study was then made of the likely benefits of water-demand pattern recognition and a method was proposed for estimating the next day's water demand. With a system solely using water-demand data for prediction and control the model is considered to be inherently linear and can be realised with classical control systems. Software was subsequently written that could sample daily demand pattern data, estimate the power spectrum and subsequently make a (linear) extrapolation into the future. The prediction figures thus obtained represent the next 24-hour hot water demand. This form of linear prediction (LP) is known to be useful and successful at extrapolating signals which are oscillatory though not necessarily periodic, and was thought to suit the nature of the domestic hot water draw-off pattern.

When the LP software algorithm was tested on simulated data it was found to only work reliably on regular and minimally varying patterns. A major disadvantage was that it was important, in the interest of accuracy, to make a correct estimation of the number of algorithm coefficients needed; unfortunately this number varied considerably from pattern to pattern, thus making it difficult, if not impossible, to synthesise a general algorithm useful for any given domestic situation.

A different method for estimating water demand was needed. At this stage it was also deemed of interest to predict this demand when faced with daily, dynamically varying electricity tariff charges; this in light of the fact that the Electricity Supply and Retail Industry is moving in this direction even on the domestic front. The combination of these factors is thought to represent a non-linear system.

When this non-linearity was added to the lack of available domestic user data (necessary for deriving a robust 'classical' prediction model) it became clear that a far more adaptable pattern recognition/forecasting model would prove necessary. Artificial Neural Networks offered a solution.

Artificial Neural Networks, or ANN for short, can be trained to form a linear or non-linear model of the underlying system. This model is then used to predict new data. Neural networks lend themselves well to pattern recognition and adaptive control; one of ANNs strong points is its ability to learn and adapt by exposing it to new data. This is considered to be important as FEMS will be operating in a non-stationary environment, i.e. it will face a different demand pattern both in time and in each domestic or industrial situation encountered.

As Neural Computing (also known as Connectionism or Parallel Distributed Processing) is a relatively new field a thorough study was made of the relevant papers and books. In the areas of 'prediction / forecasting' and 'real time systems' relatively few papers have been published with relation to Energy Management.

This reinforced the relevance of the research being done from a PhD point of view and showed that any commercial FEMS system developed could be considered to incorporate cutting edge technology.

The combination of an adaptive neural network-based prediction algorithm with the already developed cylinder control and data-acquisition system and associated hardware formed the

'intelligent' automated energy management system that was judged as being desirable for the residential, industrial and commercial customer.

Discrete time-series prediction offers a variety of neural networks and training methodologies, as evidenced by the number of books, papers and articles published on this area of application. The neural network finally selected for the energy management system was an Elman recurrent neural network; chosen because of a proven track-record, its simplicity, a capacity to reduce the noise level in data measurements without explicit knowledge of the non-linear dynamics of the system, and, most importantly, an ability to dynamically process temporal data.

At this stage the infant FEMS software was altered to incorporate a very flexible version of the recurrent network. The flexibility was needed to allow a series of tests that arrived at two usable neural net structures; one for predicting hot water demand and the other for the time of first draw-off. Thus, a full-fledged version of FEMS had been arrived at.

The FEMS system is designed to be practical, and uses the existing two-level (static) tariff rates in general use throughout New Zealand; with a cheap rate between the hours of 11 pm and 7 am. This allows the FEMS to be installed and tested in existing domestic premises. This system needs to adapt itself to the water pattern use of whichever house it is installed and reliably predict both the hot water use plus the time of first draw-off, in the next 24 hours. Being able to install it in a genuine consumer situation also eases the development of a real product that can be marketed if proven practical and economical.

Finally, an enhanced version of FEMS was considered. New Zealand has undergone some far-reaching changes in its electricity pricing structure with the introduction of a wholesale electricity market. Local electricity retail companies, such as Southpower in Christchurch, could consider introducing a variable tariff system, which will reflect in its spot price (dynamic tariff) the demand of electricity on a half-hour or hourly basis. The lower tariff prices will therefore be available in different, and probably varying, time slots during a 24-hour period.

Consumers will find it increasingly difficult to keep track of economically suitable spot-price time allotments and some form of automation in switching the hot water cylinder on and off would be both beneficial and profitable.

The second version is theoretical and centres on a discussion of the (near) future with the utilisation of a 5 or 6 level (dynamic or static) tariff rate system. This enhanced FEMS, (eFEMS), needs to not only predict how much hot water will be used but also when. If dynamic rather than static spot price tariffs are used than the varying tariffs, with their associated time periods, will also need to be forecast for the forthcoming 24 hours.

Having not two but instead up to possibly 96 predictions, which need to be matched against each other for optimal economy, it is clear that the second version poses some interesting problems. Examined amongst others points, albeit not in great depth, is the reliability of such large numbers of predictions, the approach to actually heating the right amount of water at the correct time, and whether to heat water regardless of the tariff when caught short. Also reviewed are the concepts of an energy market and the response of the small consumer.

10.1 Future research – FEMS simulation testing

The *Fluid Energy Management System*, as designed for the thesis, needs to undergo some thorough system tests as the next stage in its development. These tests would meet a number of objectives:

Substantiate the energy savings achievable with different consumer profiles.

Establish system reliability.

Determine overall system performance and eliminate bugs.

Provide consumer feedback on the ease of use.

Establish useful features for the GUI.

The system being what it is, it would be necessary to select a suitably *large* group of domestic consumers (with a *variety* of demand profiles) and convince them that it would be to their advantage to have a trial FEMS installed on their premises (commercial and industrial clients are best considered for separate trials). The cost of the various systems and the time involved in setting this up would be substantial. Points to consider are:

- The set-up, quantity and demographics of the trial installations.
- The *remote* monitoring of all test facilities (this would be desirable).
- For each site demand data would need to be collected over a *minimum of a year* to ensure seasonal influences get taken into account.
- With the data collection phase completed, the FEMS system proper would run for *another year* on each trial location.
- Both sets of results could then be compared and conclusions made.
- For ease of installation and to reduce the hardware cost it would be preferable to replace the PC with a small microprocessor, or maybe even look into the feasibility of using FPGA's (for a hardware version of the RNN).

All in all this is quite an exercise. There are, however, potentially a number of less expensive and time-consuming ways of being able to satisfy at least the first three points. One proven methodology is used (amongst others) in the car and aerospace industry and goes by the acronym of HILS, *Hardware in the Loop Simulator.*

It makes use of computer simulations to mimic hardware components. It tests, for example, the safety critical aspects of the various flight control systems.

A HILS version for FEMS could be seen as being a '*hot water cylinder and sensor real-time simulation program*'. It would be designed to fool the FEMS system into thinking that it is connected to a real domestic hot water system.

The simulation program would use *actual demand* data for the simulation's output. A comment here is that this raises a potential problem, as at present there is not enough actual demand data available to represent a large enough variation in possible installation environments.

To overcome this, the available data could be statistically modified to create additional demand patterns, although exactly how this is done would need to be investigated carefully, as it might affect the ability of the RNN to form a proper predictive model.

The dual advantage of such a HILS simulation platform is that *both* the FEMS PC version and any subsequent FEMS microprocessor versions can be thoroughly tested with the very same simulator.

On a note of caution it should be realised that interfacing the simulator to the system under test can pose some special challenges. Most I/O boards, such as the PCL-814 used in the PC version, are designed either to read sensor signals or to output actuator drive signals. In this case the situation is reversed and the real-time simulation must output sensor signals (e.g. water and ambient temp. by means of resistances to simulate temperature sensors) and read actuator drive signals (e.g. heating element on/off). In addition, if the simulation model is to be thorough, it should also incorporate failure conditions such as an unexpected power-cut or a heating element failure.

So what else is needed for a FEMS-HILS? Once a simulation is up and running in real-time it is important to be able to monitor and control it's behaviour, this will involve recording data or changing parameters whilst the simulation is in progress. An article in the *IEE Review* by D.Maclay (May 1997) suggest that a complete set of software tools (as applicable to the FEMS-HILS) should include the following:

- a 'software oscilloscope' which allows acquisition of simulation inputs, outputs or internal variables,
- selection and acquisition of signals must be possible without any interruption of the real-time simulation,
- a 'control panel' which gives a graphical display of real-time simulation parameters as well as input controls to provide two-way interaction with the simulation,
- automatic generation of real-time code; an offline simulation package such as MATLAB's SIMULINK can be used initially to develop the simulation and is followed by automatic code generation which translates the block diagram description to a software program.

It seems prudent from the above descriptions that the software tools should be carefully selected. Nonetheless, this method of hardware simulation appears a feasible alternative to a full-blown testing program and should result in reduced development time and cost.

10.2 Additional research suggestions

What follows is a list with a short description of the additional areas where the FEMS could benefit from further study. It is by no means exhaustive and would undoubtedly grow as time progressed.

The next exploratory phases

Increased data - When consideration is given to the problem of forecasting a time series with potential for wide divergence, than it becomes important to search for the best prediction model that can be designed. Collecting data from a wider variety of sources is a good first step in determining the – undoubtedly – varying nature of the time series.

System validation – Much along the lines of what was already mentioned in section 10.1, the viability of a system such as FEMS rests largely on its ability to forecast demand in not only the domestic installations but also commercial and industrial facilities. It is therefore essential to seek willing and amiable enterprises who are prepared to (i) provide test data and (ii) allow the installation and monitoring of a number of FEMS trial systems in their plant operating

environments. It also provides much needed feedback on the acceptance of FEMS by potential customers.

Alternative sensors – The thermistor temperature sensors used in FEMS, although cheap and reliable, are not optimal in terms of accuracy, installation and operating characteristics. Another disadvantage is that support electronics are needed to make them function. It is worthwhile therefore, to investigate the alternatives that might have arrived recently on the market and fit in better with the image of FEMS as a sophisticated intelligent system.

Customer interface panel – The test systems as covered in this thesis made use of software GUIs displayed on, and operated by, monitor and keyboard respectively. Although on a considerably smaller scale, a hardware version of FEMS will need similar attributes. What needs to be investigated is the form and function of a display and key panel (see also *Appendix* C).

Software streamlining – In its present form the various modules that make up the software side of FEMS posses additional lines of code; this allows the programmer to cover and monitor every eventuality as well as facilitates debugging. Prior to scaling the system down to a suitable microprocessor (see the next page) this extraneous coding needs to be removed and a critical look needs to be taken at improving (faster, shorter, simpler) the remaining modules.

Customer acceptance – To date FEMS is a concept that is considered a 'good idea' by a very small group of well-educated people. An actual system would need substantial market research input to ensure that the population at large is willing to accept (and ultimately purchase) such energy-saving equipment, given the right price and profile.

Alternatives to RNN

An interesting paper by Mozer (1993) has very recently been obtained from the internet, which questions the adequacy of recurrent neural net architectures for difficult temporal processing and prediction tasks, and advocates the use of TDNN networks as inherently providing superior predictions. Mozer also offers a number of alternative forms of recurrent networks that might be worth exploring. This needs to be followed up.

Communicating tariffs to eFEMS

A recent arrival on the local high-tech industry scene, Indranet, has offered an excellent means of communicating varying electricity tariffs to equipment that can take advantage of these prices. Indranet intends to fabricate a cheap 'black box' transmitter/receiver with a wide bandwidth and high rate of transmission. Energy management systems will no doubt reap benefits from this type of communication device; eFEMS certainly could.

Rapid recovery cylinders

Improvement in the design and construction of hot water cylinders is another direction where research could make a difference. Rapid recovery systems do exist, albeit in small numbers, and usually take the form of either a second heating element at the top, or an inner container of water around the element that separates it from the surrounding water (see the *Appendixes*). The latter design especially, would augment the overall efficiency factor obtained with a FEMS system due to increased stratification within the cylinder and being able to heat water closer to the time of actual use.

Scaling down to a microprocessor

The present state of FEMS is a collection of hardware boards plus a PC running relatively complex software. The need to scale all this paraphernalia down to a cheap 16 bit or 32 bit microprocessor is a must if the system is ever to find a suitable niche in the after-market section of energy efficient equipment. In order to get an idea of what was involved a project proposal, originally intended for a third professional year electrical engineering student, is shown in *Appendix* C. To date there have been no takers.

Chaos

Some thought should be given to possible chaotic components in the data. The presence of these can be tested for (see Kim et al., 1993). Chaos is inherently non-linear and a chaotic time series has a broadband spectrum, but so unfortunately does a random time series. This means that conventional measures such as auto-correlation or Fourier power spectra are inadequate to describe it (is it chaos or is it noise?). Chaos provides a link between deterministic systems and random processes, with both good and bad implications for the prediction problem. In a deterministic system, chaotic dynamics can amplify small differences, which in the long run produces effectively unpredictable behaviour. On the other hand, chaos implies that not all random-looking behaviour is the product of complicated physics. Under the influence of non-linearity, only a few degrees of freedom are necessary to generate chaotic motion. In this case, there is potential to model the behaviour deterministically and to make short-term predictions possible (Eubank et al., 1990). Chaos is thus a double-edged sword; it implies that even approximate long-term predictions may be impossible, but that very accurate short-term predictions may be possible. Food for thought.

Post Scriptum

Triple Function Heat Pumps (TFHP)

One of the external examiners, Professor Gerald T. Heydt of Arizona State University, commented that he would have liked to have seen the TFHP being mentioned in the thesis as it apparently represents a relatively well known, though infrequently used, way of heating hot water in the US/Canada region.

To my mind it is questionable whether the TFHP is in keeping with the envisaged aim of the thesis, which is on the predictive control of fluid heating (using the domestic water heater as an example) with a bias towards the New Zealand situation. If the TFHP is included, than why not other useful methods of heating such as solar, wetback, induction, etc.? It is nonetheless an interesting concept for a multipurpose heating/cooling system. If seen in the larger context (which is undoubtedly what Prof. Heydt has in mind) the idea behind FEMS could be applied in conjunction with the TFHP to produce an even more efficient HW heating system.

Information on the TFHP is not in wide circulation. A literature search in the library produced no results; he WWW fared slightly better (3 hits) but details were missing (see extract below) or the URL no longer existed; the websites for Carrier and Trane made no mention of the TFHP.

[*Extract*: The Triple Function Heat Pump, developed by Mississippi Power Company, represents new technology in the area of heat pumps integrated with demand water heating. The unit has *four* operating modes: standard space cooling, space heating, simultaneous space cooling and water heating, and demand water heating or dedicated water heating. A 70% reduction in energy use for water heating is achieved by distributing geothermal heat through an earth-coupled, closed-loop heat pump system or from indoor air.]

In fact most articles and reviews concentrate on the *dual* function type of Heat Pump Water Heater. The HPWH uses the reverse refrigeration cycle to *remove* heat from ambient air, transferring this heat to the water in the cylinder while simultaneously acting as an air-conditioner. In this situation it is *2 to 3 times more efficient* than resistive hot water heating.

A recent report from the American Federal EREN (Energy Efficiency and Renewable Energy Network) provides insight into what the possible problems are with the HPWH (and by extension the TFHP) in gaining acceptance in the marketplace.

Basically the feedback on the use of the HPWH is very *positive*. Consumer acceptance seems to be more a problem of getting the right information through to the potential customers. The EREN report specifically mentions that the market in HPWHs could grow if:

Information is provided on successful installations.

Information is provided on available technology.

Installation guidelines are provided for facility managers/building contractors.

Additionally the report states that only a small number of companies manufacture HPWHs, and even than the equipment is mainly intended for *larger* premises such as office buildings. The reason for this is that the HPWH becomes increasingly attractive in building applications where the energy costs are high, and where there is a *steady* demand for hot

water i.e. it is not so much a function of the *type* of building but rather *demand* and energy *cost*. Presumably, with relatively large machines and low manufacturing volumes, this keeps the purchase price too high to make it attractive to the domestic consumer.

The EREN report does specifically mention the triple function HPWH, but only in the same breath as the 'standard' HPWH and does not elude on any problems faced by this design.

A recent paper by Shelton et al. (1999) highlights efforts made by the U.S. DoE to pass a mandate that would force all new homes to install heat pumps instead of electric powered water heating. This failed; the response from utilities and manufacturers was strongly *negative*. Electric utilities claimed that there was an inadequate infrastructure of technicians who could install and service electric heat HPWH. They also stated that the initial financial burden was too high for may consumers, and that increasing the efficiency of appliances in the home, such as water heaters, would reduce the incentive to improve plant efficiency (Pacific Northwest National Laboratory, 1998).

Are the disadvantages of cost and maintenance for the TFHP sufficiently serious to limit widespread adoption? To my way of thinking the reply should be *no*; but it needs some work doing to improve that situation. It can be argued that the manufacturers could overcome the cost and maintenance problems of the TFHP (Carrier Corporation is after all the world's largest manufacturer of HVAC, their budget should stretch to this). Adjustable speed drive technology is well developed and maybe the manufacturers should look to the car industry (DAF and others) for comparable technology. They should couple this to a concerted information and marketing drive, targeting facility managers first and residential consumers second. Concurrently they should offer training courses for installation technicians. A gradual introduction of these measures should see the HWHP, in its various forms, gaining wider acceptance.

Post Scriptum References:

EREN - Energy Efficiency and Renewable Energy Network, (199?), Commercial heat pump water heaters, http://www.eren.doe.gov/femp/prodtech/10_comm.html

Shelton, S.V., Schaefer, L.A., (1999), The economic payoff for global warming emissions reduction, *internal report* for the G.W.W. school of Mechanical Engineering, Georgia Institute of Technology, Atlanta, U.S.A., http://www.me.gatetech.edu/energy.

Pacific Northwest National Laboratory, (1998), Technology assessment and screening analysis, *Appendix B, supplement to the water heater rulemaking framework*, DoE/OCS, Richland, Washington, USA, p18-19.

References

ACC and PLUNKET, (1990), Hot water burns like fire, *Standards Magazine*, Vol.40, No.10, Wellington, New Zealand.

ACC, (1990), Publicity Brochures, Accident Compensation Corporation, Wellington, New Zealand.

Ackley, D.H., Hinton, G.E., Sejnowski, T.J., (1985), A learning algorithm for Boltzmann machines, *Cognitive Science*, 9, 147 –169.

Advantech Co. Ltd, (1992), *PCL-814 Modulized DAS card*: user's manual.

Allera, S.V., Cook, A.A., (1994), *Domestic customer response to a multi-rate tariff*, internal report for The Electricity Association, U.K.

Allis, L. (1994), *Searching for Solutions in Games and Artificial Intelligence*, PhD thesis, University of Limburg.

Al-Marafie, A., Moustafa, S.M., and Al-Kandarie, A., (1989), Factors affecting static stratification of thermal water storage, *Energy Sources*, Vol.11, No.3.

Altman, E., Koole, G., (1993), Stochastic scheduling games with Markov decision arrival processes, *Computers and Mathematics with Applications*, Vol.26, No.6, 141-148.

Anderson, C, (1989), Learning to control an inverted pendulum using neural networks, *IEEE Control Systems Magazine*, April, 31 - 36.

Anderson, J.A., Rosenfeld, E., (1989), *Neuro-computing: foundations of research*, MIT Press, Cambridge, MA.

Andre, E.R., (1988), *The effect of spot pricing electricity and remote metering on power system development*, Riccarton Borough Electricity Supply, Christchurch, New Zealand.

Andreou, A.G., (1992), *Minimal circuit models of neurons, synapses and multivariable functions for analog VLSI neuromorphic computation*, Report JHU/ECE-92-13, Johns Hopkins University, Baltimore, MD.

Åström, K.J., Bohlin, T., (1965), Numerical identification of linear dynamic systems from normal operating records, *IFAC Symposium on Self-adaptive Systems*, Teddington, England.

Åström, McAvoy, (1992), *Intelligent control: an overview and evaluation*, White, D.A., Sofge, D.A., (eds.), Van Nostrand- Reinhold, New York.

Baldi, P., (1988), Neural networks, orientations of the hypercube and algebraic threshold functions, *IEEE Trans. on Information Theory*, Vol. IT-34, 523 - 530.

Baldi, P., Atiya, A., (1994), *How delay affects neural dynamics and learning*, internal report, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. (see also ftp://archive.cis.ohio-state.edu/pub/neuroprose/baldi.delays.ps.Z).

Bartlett, P.L., (1997), For valid generalisation, the size of the weights is more important than the size of the network, in *Advances in Neural Information Processing Systems* 9, Mozer, M.C., Jordan, M.I., and Petsche, T., (eds.), Cambridge, MA: The MIT Press, 131 -140.

Barto, A., Sutton, R., Anderson C., (1983), Neuron-like adaptive elements that can solve difficult learning problems, *IEEE Transaction on Systems, Man, and Cybernetics*, 13, 834 - 846.

Baum, E.B., (1991), Neural net algorithms that learn in polynomial time from examples and queries, *IEEE Transactions on Neural Networks*, 2, 5 - 19.

Beale, R., Jackson, T., (1990), *Neural computing: an introduction*, Adam Hilger.

Beer, R.D., (1994), On the dynamics of small continuous-time recurrent networks, *technical report* CES-94-18, Case Western University, Cleveland, OH.

Bishop, C.M., (1995), *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford.

Bishop, C.M., Svensen, M., Williams, C.K.I., (1997), GTM: a principled alternative to the self-organising map, in *Neural information processing systems* 9, MIT press, Cambridge, MA. (also website http://www.ncrg.aston.ac.uk/GTM/).

Blum, E., Li, L., (1991), Approximation theory and feedforward networks, *Neural Networks*, 4, 455 - 463.

Blum, E.K., Wang, X., (1992), Stability of fixed points and periodic orbits and bifurcations in analog neural networks, *Neural Networks*, 5, 577 - 587.

Bodenhausen, U., Waibel, A.H., (1991), Learning the architecture of neural networks for speech recognition, in *Proceedings ICASSP*.

Bohn, R.E., Golub, B.W., Tabors, R.D., Schweppe, F.C., (1984), Deregulating the generation of electricity through the creation of spot markets for bulk power, *The Energy Journal*, Vol.5, No.2, 71-91.

Boser, B.E., Säckinger, E., (1992), Hardware requirements for neural network pattern classifiers, *IEEE Micro* 12, 32 - 40.

Bowling, T., (1992), *An adaptive control system for a domestic hot water system*, unpublished Final Year Project, Dept. of Electrical Engineering, University of Canterbury.

Box, G.E.P., Jenkins, G.M., (1970), *Time series analysis, forecasting and control*, San Francisco, Holden Day.

Brigham, E.O., (1974), *The Fast Fourier Transform*, Prentice-Hall.

Brooks, F.P., (1987), No silver bullet: accidents and essence of software engineering, *IEEE Computer*, 10 - 19.

Broomhead, D.S., Lowe, D., (1988), Multivariable functional interpolation and adaptive networks, *Complex Systems*, 2, 321 - 255.

Brown, C.N., (1985), Charging for electricity in the early years of electricity supply, *IEE Proceedings*, Vol.132, Pt. A, No.8.

Brown, R.G., (1963), *Smoothing, forecasting and prediction of discrete time series*, Prentice-Hall.

Busch, F.J., Eto, J., (1996), Estimation of avoided cost for electric utility demand side planning, *Energy Sources*, 18, 473 – 499.

CAE, (1996), *Energy efficiency: a guide to current and emerging technologies*, Centre for Advanced Engineering, Vol.1, University of Canterbury, Christchurch, New Zealand.

Canu, S., (1990), Formal Neural Network as an Adaptive Model for Water Demand, *International Neural Network Conference*, Vol.1, Paris, Kluwer Academic Publishers.

Casey, M.P., (1995), Computation in discrete-time dynamical systems, *PhD thesis*, Department of Mathematics, University of California, San Diego.

Casey, M.P., (1995b), Relaxing the symmetric weight condition for convergent dynamics in discrete time recurrent networks, *technical report* INC-904, Institute for Neural Computation, University of California, San Diego.

Caudill, M., (1991), Neural network training tips and techniques, *AI Expert*, January, 56 - 61.

Caudill, M., (1991b), Avoiding the great back-propagation trap, *AI Expert*, July, 29 - 35.

Champeney, D.C., (1973), *Fourier transforms and their physical applications*, New York, Academic Press.

Chao, H., Peck, S., (1997), An institutional design for an electricity contract market with central dispatch, *Energy Journal*, Vol.18, No.1, 85 – 110.

Chen, D., Giles, C., Sun, G., Chen, H., Less, Y., Goudreau, M. (1993). Constructive learning of recurrent neural networks, *IEEE Int. Conf. on Neural Networks*, 3, 1196 – 1201.

Childers, D.G., (1978), *Modern spectrum analysis*, IEEE Press, New York.

Cichosz, P., (1995), *Truncating Temporal Differences: On the Efficient Implementation of TD(λ) for Reinforcement Learning*, Vol. 2, 287-318. (see also website ftp.mrg.dist.unige.it directory: pub/jair/pub/volume2).

Cichosz, P., Mulawka, J., (1995), Fast and efficient reinforcement learning with truncated temporal differences, In *Proc. of the Twelfth Int. Conf. on Machine Learning*, Morgan Kaufmann.

Cohen, M.A., Grossberg, S., (1983), Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-13, 815 - 826.

Cottrell, G., Munro, P., Zipser, D., (1987), Image compression by back-propagation: an example of extensional programming, *ICS report* 8702, University of California, San Diego.

Dauncey, G. (1990), The role of new metering technologies in combating the greenhouse effect, *IEE 6th International Conference on Metering Apparatus and Tariffs for Electricity Supply,* Manchester, 57 - 61.

Dayan, P, (1992), The convergence of TD(λ) for general λ, *Machine Learning* 8, Kluwer Academic Publishers, Boston, 341 - 362.

Dayan, P, (1994), TD(λ) converges with probability 1, *Machine Learning* 14, Kluwer Academic Publishers, Boston, 295 - 301.

DeMarco, T., (1978), *Structured analysis and system specification*, Yourdon Press, New York.

Dempo, A., Farotimi, 0., Kailath, T., (1991), High-order absolutely stable neural networks, *IEEE Trans. on Circuits and Systems*, Vol.38, No.1.

Demuth, H., Beale, M., (1995), *Neural network toolbox*, The Math Works Inc.

Devroye, L., Gyorfi, L., Lugosi, G., (1996), *A Probabilistic Theory of Pattern Recognition*, Springer, NY.

Dick, A.J., Allera, S.V., Horsburgh, A.G., (1990), EMU – the energy management unit, *IEE 6th Int. Conference on Metering Apparatus and Tariffs for Electricity Supply*, Manchester, England, 177 – 182.

Dingle, A.A., (1992), *Engineering in brain research - Processing electro-encephalograms and chaos in neural networks*, unpublished PhD thesis, Canterbury University Press, New Zealand.

Doyle, K.M., (1990), *Plumbing and Gasfitting*, Vol.2 – Services and Roofing, Plumbing, Gas & Drainlaying Foundation.

Eberhart, R.C., Dobbins, R.W., (1990), *Neural network PC tools: a practical guide*, Academic Press.

EECA, (1991), *Reduce Electric Hot Water Heating Costs with a CylinderWrap*, Project Summary 25, Energy Efficiency and Conservation Authority, Wellington, New Zealand.

EECA, (1994), *Supply curves and the economic potential for improving energy efficiency in New Zealand*, Energy Efficiency and Conservation Authority, Wellington, New Zealand.

EECA, (1995), *Making the most of your hot water system*, Energy Efficiency and Conservation Authority, Wellington.

Electricity Association, (1997), *Building the electricity market place*, website: http//www.electricity.org.uk/inds_fr.html.

Electricity Association, (1998), *Load profiles in the 1998 electricity market place*, website: http//www.electricity.org.uk/inds_fr.html.

Elliot, D.F., Rao, K.R., (1982), *Fast Transforms: algorithms, analyses, applications, New York*, Academic Press.

Elman, J.L., (1990), Finding structure in time, *Cognitive Science*, 14, 179 - 211.

Elsner, J.B., (1992), Predicting time-series using a neural network as a method for distinguishing chaos from noise, *Journal of Physics A* 25, 843 - 850.

Encarta, Microsoft, (version 1997), *World Atlas*.

Erwin, E. Obermayer, K., Schulten, K., (1995), *Neural networks for pattern recognition*, Oxford University press.

Eubank, S., Farmer, J.D., (1990), An introduction to chaos and prediction, in *Proceedings of the Santa Fe Instit. Summer school*, E.Jen (ed), Addison-Wesley, Reading, MA, 75 – 190.

Evans, M., (1984), *Computer modelling of New Zealand domestic hot water systems*, Final Year Project, Dept. of Mechanical Engineering, University of Canterbury, New Zealand.

Faggin, F., (1991), VLSI implementation of neural networks, Tutorial Notes, *International Joint Conference on Neural Networks*, Seattle, WA.

Fakhr, W., Kamel, M., Elmasry, M.I., (1992), Probability of error, maximum mutual information, and size minimisation of neural networks, *Int. Joint Conf. On Neural Networks*, Baltimore, MD.

Fidgett, J., Gray, F.M., Mahoney, A.P., (1987), Radio teleswitching tariff and load management system, *IEE 5th International Conference on Metering Apparatus and Tariffs for Electricity Supply*, Edinburgh, 272 - 276.

Foord, P., Tsoucalas, J., (1990), Remote meter reading, load control and distribution system automation utilising SWD technology, *IEE 6th Int. Conference on Metering Apparatus and Tariffs for Electricity Supply*, Manchester, 163 - 167.

Forrest, A., Hoskins, J., (1982), Operating experience with modern mains signalling equipment in the states of Guernsey Electricity Board, *IEE 4th International Conference on Metering Apparatus and Tariffs for Electricity Supply*, London, 52 - 55.

Frasconi, P., Gori, M., (1996), Computational capabilities of local-feedback recurrent networks acting as finite-state machines, *IEEE Transactions on Neural Networks*, Vol.7, No.6, 1521 - 1524.

Frasconi, P., Gori, M., Soda G., (1992), Local feedback multilayered networks, *Neural Computation*, Vol.4, No.1, 120 - 130.

Fukushima, K., (1980), Neocognitron: a self-organising neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, 36, 193-202.

Geller, H., Nadel, S., Pye, M., (1999), *Demand-side management at a crossroads*, DA/DSM Europe 96.

Geman, S., Bienenstock, E., Doursat, R., (1992), Neural Networks and the Bias/Variance Dilemma, *Neural Computation*, 4, 1 - 58.

Ghosh, K., Ramesh, V.C., (1997), An options model for electric power markets, *Electrical Power and Energy Systems*, Vol.19, No.2, 75 – 85.

Giles, C. L., Chen, D., Miller, C.B., Chen, H.H., Sun, G.Z., Lee, Y. C., (1991), Second-order recurrent neural networks for grammatical inference, *Proc. Int. Joint Conf. on Neural Networks, IJCNN91*, vol. II., 273-281. (see also http://www.neci.nec.com/~giles/papers/).

Ginzberg, I., Horn, D., (1991), Learnability of time series, *Int. Joint Conf. on Neural Networks*, Singapore, 2653 - 2657.

Golub, B.W., Bohn, R.E., Tabors, R.D., Schweppe, F.C., (1983), *An approach for deregulating the generation of electricity*, ch.5 in Plummer, editor.

Gorman, R.P., Sejnowski, T.J., (1988), Learned classification of sonar targets using a massively parallel network, *IEEE Trans. on Acoustics, Speech and Signal Processing* 36, 1135 - 1140.

Gustafson, M.W., Baylor, J.S., Epstein, G., (1993), Direct water heating load control – estimating program effectiveness using an engineering model, *IEEE Trans. On Power Systems*, Vol.8, No.1, 137 – 143.

Haffner, P., Waibel, A., Sawai, H., Shikano, K., (1989), Fast back-propagation learning methods for large phonemic neural networks, *Proceedings of the EuroSpeech Conference*.

Haines, R., (1987), Control Systems for Heating, Ventilating and Air Conditioning, 4th edn, Van Nostrand Reinhold, New York.

Hammerstrom, D., (1992), Electronic neural network implementation, Tutorial No.5, *International Joint Conference on Neural Networks*, Baltimore, MD.

Harmon, M., Baird, L. (1995), Residual advantage learning applied to a differential game. *Neural Information Processing Systems* 7.

Harris, G., (1993), *Promoting the market for energy efficiency*, report to the Officials Committee on Energy Policy, Wellington, New Zealand.

Harrison, P.J., Stevens, C.F., (1976), Bayesian forecasting, *Journal Statistical Soc.*B, 38.

Harvey, A.C., (1981), *Time series models*, New York, Wiley.

Haykin, S., (1994), *Neural networks: A comprehensive foundation*, Macmillan.

Hebb, D.O., (1949), *The organization of behavior: a neurophysiological theory*, New York, Wiley.

Hecht-Nielsen, R., (1991), *Neurocomputing*, Addison-Wesley, Reading, MA.

Hecht-Nielsen, R., (1992), Theory of the back-propagation network, in *Neural Networks for Perception, Vol.2*, Harry Wechsler (ed), Academic Press, New York.

Hendtlass, C.A., (1981), *Report on a survey into hot water usage patterns in residential dwellings*, Report of Joint Centre for Environmental Sciences, University of Canterbury, Christchurch, New Zealand.

Henny, A, (1987), *Privatise power: restructuring the electricity supply industry*, Centre for Policy Studies, Studie no.83, London.

Hoeseinen, B., Mead, C.A., (1972), Fundamental limitations in microelectronics. I. MOS technology, *Solid-state electronics* 15, 819 - 829.

Holland, J.H., (1986), Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems, In *Machine Learning: An Artificial Intelligence Approach*, Vol.2, S.Michalski, J.G.Carbonell, T.M.Mitchell (eds), Morgan Kaufmann.

Holt, C.C., (1957), Forecasting trends and seasonals by exponentially weighted moving averages, *O.N.R. memorandum No.52*, Carnegie Institute of Technology.

Hopfield, J.J., (1982), Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences*, 79, 2554 - 2558.

Hopfield, J.J. (1984), Neurons with graded response have collective computational properties like those of two-state neurons, *Proceedings of the National Academy of Sciences,* 81, 3088 - 3092.

Hornik, K., (1991), Approximation capabilities of multilayer feedforward networks, *Neural Networks*, Vol.4, No.3.

Hornik, K., Stinchcombe, M., White H., (1989), Multilayer feedforward networks are universal approximators, *Neural Networks*, Vol.2, No.5, 359 - 366.

Horowitz, P., (1981), *The art of electronics*, Cambridge University Press.

Hui, S., Zak, S.H., (1992), Dynamical analysis of the brain-state-in-a-box neural models, *IEEE Transactions on Neural Networks*, No.1, 86 - 94.

Hummels, D.W., Ahmed, W., Musavi, M.T., (1995), Adaptive detection of small sinusoidal signals in non-Gaussian noise using an RBF neural network, *IEEE Transactions on Neural Networks*, Vol.6, No.1.

Hunt, E.B., (1975), *Artificial Intelligence*, Academic Press.

Hunt, K. J., Sbarbaro, D., (1991), Neural networks for non-linear internal model control, *IEE Proc.-D*, Vol.138, No.5, 431 - 438.

Hunt, K. J., Sbarbaro, D., Zbikowski, R., Gawthrop, P. J., (1992), Neural networks for control systems - a survey, *Automatica*, Vol.28, No.6, 1083 - 1112.

Hunt, S., (1995), *The design of a domestic energy management system using a distributed control network*, unpublished Masters of Electrical Engineering Thesis, University of Canterbury, New Zealand.

Iglewicz, B., (1983), Robust scale estimators and confidence intervals for location, in *Understanding Robust and Exploratory Data Analysis*, Hoaglin, D.C., Mosteller, M. and Tukey, J.W., eds, Wiley, New York.

Jordan, M., (1986), Attractor dynamics and parallelism in a connectionist sequential machine, *Proceedings of the 8th annual conference of the Cognitive Society.*

Jury, E.I., (1964), *Theory and application of the z-transform method*, Wiley, New York.

Kalman, B.L., and Kwasny, S.C., (1992), Why tanh? Choosing a sigmoidal function, *International Joint Conference on Neural Networks*, Baltimore, MD.

Kalman, R.E., (1960), A new approach to linear filtering and prediction problems, *Trans.ASME*, Journal of Basic Engineering, 82.

Karjala, T.W., Himmelblau, D.M., (1992), Data rectification using recurrent (Elman) neural networks, *IEEE Trans. on Neural Networks*, Vol.2, 901 - 906.

Karnin, E.D., (1990), A simple procedure for pruning back-propagation trained neural networks, *IEEE Trans. on Neural Networks*, Vol.1, No.2, 288 - 291.

Kenue, S.K., (1991), Efficient activation functions for the backpropagation neural network, *SPIE, Proceedings from Intelligent Robots and Computer Vision X: Neural, Biological, and 3-D Methods*, (November).

Keyes, R.W., (1987), *The physics of VLSI systems*, Addison-Wesley, Reading, MA.

Kim, H.H., Stringer, J, (1993), *Applied chaos*, John Wiley, New York.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., (1983), Optimization by simulated annealing, *Science*, 220.

Kohonen, T., (1988), *Self-organization and associative memory*, Springer Verlag

Kohonen, T., (1990), The self-organizing map, *Proceedings of the IEEE*, 78, 1464 - 1480.

Kohonen, T, (1995), *Self-organising maps*, Springer-Verlag, Berlin.

Kohonen, T., (1982), Self-organised formation of topologically correct feature maps, *Biological Cybernetics*, 43, 59 - 69.

Kosmatopoulos, E.B., Polycarpou, M.M., Christodoulou, M.A., Ioannou P.A., (1995), High-Order Neural Network Structures for Identification of Dynamical Systems, *IEEE Trans. on Neural Networks*, Vol.6, No.2, 422 - 431.

Kreyszig, E., (1983), *Advanced engineering mathematics*, Wiley.

Lütkepohl, H., (1993), *Introduction to multiple time series analysis*, Springer-Verlag.

Lang, K.J., Waibel, A.H., Hinton, G.E., (1992), A time-delay neural network architecture for isolated word recognition, in *Artificial neural networks*, E. Sanchez-Sinencio, C. Lau, (eds.), IEEE Press, New York, 388 – 408.

Larimore, W., (1983), System identification, reduced order filtering and modelling via canonical variate analysis, *Proc. American Control Conference*, San Francisco.

Lavan, Z., Thompson, J., (1977), Experimental study of thermally stratified hot water storage tanks, *Solar Energy, V19*.

Lawrence, S., Chung Tsoi, A., Giles, C.L., (1997), Noisy time series prediction using symbolic representation and recurrent neural network grammatical inference, *Technical Report* UMIACS-TR-96-27 *and* CS-TR-3625, Institute for Advanced Computer Studies, University of Maryland.

Lawrence, S., Giles, C.L., Fong, S., (1996), On the applicability of neural network and machine learning methodologies to natural language processing, appears in: Workshop on New Approaches to Learning for Natural Language Processing, *Int. Joint Conf. on Artificial Intelligence*, Montreal, Canada, 1–8.

Lawrence, S., Giles, C.L., Fong, S., (1999), Natural language grammatical inference with recurrent neural networks, *IEEE Transactions on Knowledge and Data Engineering, accepted for publication.*

Le Cun, Y., Jackel, L.D., Boser, B., Denker, J.S., Graf, H.P., Guyon, I., Henderson, D. Howard, R.E., Hubbard, W., (1992), Hand-written digit recognition: applications of neural network chips and automatic learning, in *Artificial Neural Networks*, IEEE Press, 463 - 468.

Lee, S.E., Bradley, R.H., (1992), Regression analysis of spectroscopic process data using a combined architecture, *IEEE Trans. on Neural Networks*.

Lefas, C.C., (1987), Microprocessor control of distributed storage, active solar heating systems, *Journal of Microcomputer Applications*, 10.

Lescoeur, B., Galland, J.B., (1986), Tariffs and load management: the French experience, paper presented at the *IEEE Power Eng. Soc. summer me*eting, Mexico City, 312-3

Levermore, G.J., (1992), *Building energy management systems, an application to heating and control*, E&FN Spon, London.

Levine, W.S., (1996), *The control handbook*, IEEE Press.

Lin, D.T., Ligomenides, P.A., Dayhoff, J.E., (1993), Learning spatio-temporal topology using an adaptive-time delay neural network, *World Congress on Neural Networks*, Vol.1, Portland, OR, 291 - 294.

Liou, R., Azimi-Sadjadi. M.R., Dent, R., (1991), Detection of dim targets in high cluttered background using high order correlation neural network, *Proc. Int. Joint Conf. on Neural Networks, IJCNN91*, Vol.I, 701 - 706.

Lippmann, R.P., (1987), Introduction to computing with neural nets, *IEEE Acoustics Speech and Signal Processing Magazine*, April, 4 - 22.

Ljung, L., (1996), *System Identification*, CRC Press Inc.

Ljung, L., Glad, T., (1994), *Modelling of dynamic systems*, Prentice-Hall.

Luenberger, (1989), *Linear and non-linear programming,* Addison Wesley, 2nd edition.

Lugosi, G., and Zeger, K. (1995), Nonparametric estimation via empirical risk minimization, *IEEE Trans. on Information Theory*, 41.

Mache, N., Reczko, M., Levi P., Hatzigeorgiou, A., (1998), *Multistate Time-Delay Neural Networks for the recognition of POL II promoter sequences*, Internal report, No.116, Institute of Parallel and Distributed High-Performance Systems, University of Stuttgart, Germany.

Mair, F.J., (1994), *The application of neural network temporal difference techniques to dynamical systems*, unpublished Master's report, Dep. Electrical and Electronic Eng., University of Auckland, Auckland, New Zealand.

Marquez, L., Hill, T., O'Connor, M. Remus, W., (1992), Neural network models for forecast: a review, in *Artificial Neural Networks*, IEEE Press, 494 - 498.

Marr, D., Poggio, T., (1976), Cooperative computation of stereo disparity, *Science*, 194, 283 - 287.

Masters, T., (1994), *Practical neural network recipes in C++*, Academic Press.

Matty, T.C., (1989), Advanced Energy Management for Home Use, *IEEE Transactions on Consumer Electronics,* Vol.35, No.3, August, 584 - 587.

McClelland, J.L., (1989), *Parallel distributed processing: implications for cognition and development*, Clarendon Press.

McCulloch, W.S., and Pitts, W.H., (1943), A logical calculus of the idea immanent in nervous activity, *Bulletin of Mathematical Biophysics* 5, 115 - 133.

McDonald, J.R., Whiting, P.A., Lo, K.L. (1994), Spot-pricing: evaluation simulation and modelling of dynamic tariff structures, *Electrical Power and Energy Systems*, Vol.16, No.1, 23 - 34.

McDonald, J.R., Whiting, P.A., Lo, K.L., (1994b), Optimized reaction of large electrical consumers in response to spot-price tariffs, *Electrical Power and Energy Systems,* Vol.16, No.1., 35 - 48.

McGlinchy, B., (1995), Metering: the road ahead, *Electrical Focus*, December/January, 60 - 64.

M-co, (1999), *Changing right before your eyes*, Market report 1998, Wellington, New Zealand.

McShane, J., (1992), An introduction to neural nets, *Hewlett-Packard Journal*, February, 62 - 65.

Meier, H., (1993), *New cost-based tariffs for low-voltage consumers: the reform of the general tariffs in the Federal Republic of Germany*, RWE Energie AG, Germany.

Minsky, M., and Papert, S., (1969), *Perceptrons*, MIT Press, Cambridge, MA.

Moody, J.E., (1992), The effective number of parameters: an analysis of generalisation and regulisation in nonlinear learning systems, in *Advances in Neural Information Processing Systems*, J.E.Moody, S.J.Hanson, R.P.Lippman, (eds.), Morgan Kaufmann, San Mateo, CA.

Mozer, M.C., (1989), A focused backpropagation algorithm for temporal pattern recognition, *Complex Systems*, Vol.3, No.4, 349 – 381.

Mozer, M.C., (1992), Induction of multiscale temporal structure, in *Advances in Neural Information Processing Systems*, J.E.Moody, S.J.Hanson, R.P.Lippman, (eds.), Morgan Kaufmann, San Mateo, CA.

Mozer, M.C., (1993), Neural net architectures for temporal sequence processing, in *Predicting the Future and Understanding The Past*, Addison-Wesley Publishing, Redwood City, CA

Mozer, M.C., Jordan, M.I., Petsche, T., (eds.), (1996), *Advances in Neural Information Processing Systems* 9, The MIT Press, Cambridge, MA.

Mynatt, B.T., (1990), *Software Engineering with Student Project Guidance*, Prentice Hall, New Jersey.

Narendra, K. S., Parthasarathy, K., (1990), Identification and control of dynamical systems using neural networks, *IEEE Trans. on Neural Networks*, Vol.1, No.1, 4 - 27.

Narendra, K.S., Parthasaranty, K., (1991), Gradient methods for the optimisation of dynamical systems containing neural networks, *IEEE Trans. on Neural Networks*, Vol.2, No.1, 252 - 262.

NBRI, (1976), *Introductory guide to solar energy and solar water heaters*, National Building Research Institute, Pretoria, South Africa.

Nerrand, O., Roussel-Ragot, P., Urbani, D., Personnaz, L., Dreyfus, G., (1994), Training recurrent neural networks: why and how? An illustration in dynamical process modelling, *IEEE Trans. on Neural Networks*, Vol.5, No.2, 178 - 184.

Nguyen, D.H., Widrow, B., (1990), Neural networks for self-learning control systems, *IEEE Control Systems Magazine*, April, 18 - 23.

Nunn, C., Moore, P.M., Williams, P.N., (1992), Remote meter reading and control using high-performance PLC communications over the low voltage and medium voltage distribution networks, *IEE 7th International Conference on Metering Apparatus and Tariffs for Electricity Supply,* Glasgow, 304 - 308.

Omlin, C. W., Giles, C. L. (1993), Pruning recurrent neural networks for improved generalization performance, *Technical Report Tech Report No 93-6*, Computer Science Department, Rensselaer Polytechnic Institute.

Oppenheim, A.V., Schafer, R.W., (1989), *Discrete-time signal processing*, Prentice-Hall.

Orr, G.B., Mueller, K.R., (eds), (1998), *Neural Networks: tricks of the trade*, Springer, Berlin.

Overschee, P.V., DeMoor, B., (1994), Subspace algorithms for the identification of combined deterministic-stochastic systems, *Automatica*, 30.

Pao, Y, (1989), *Adaptive pattern recognition and neural networks*, Addison-Wesley, Reading, MA.

Paretto, P., Niez, J.J., (1986), Long term memory storage capacity of multiconnected neural networks, *Biology and Cybernetics*, Vol. 54, 53 - 63.

Parker, G.J., (1982), *Results from a computer simulation of a domestic hot water system*, ECRC Memorandum M1560, Electricity Council Research Centre, UK.

Parker, G.J., (1993), Off-peak energy storage for domestic applications in Christchurch, New Zealand, *Applied Energy* 44, 259 - 281.

Parker, G.J., Tucker, A.S., (1991), Dynamic simulation of a domestic hot water system, *Applied Energy* 40, 1 - 19.

Philips Co., (1984), *The Philips data handbook - Components and materials*, part 11.

Pinenda, F.J., Generalisation of back-propagation to recurrent networks, *Phys.Rev.Letters*, Vol.59, No.19, 2229 - 2232.

Pitkin, E.T., (1979), Solar plus waste-heat recovery hybrid water-heating system, *Int. Congress International Solar Energy Society*, Atlanta, Georgia, Pergamon Press, Oxford.

Potvin, J.Y., (1993), The travelling salesman problem; a neural network perspective, *Journal of Computing*, 5, 328 – 437.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., (1992*), Numerical recipes in C, the art of scientific computing*, Wiley.

Rattray, W., McDonald, J.R., (1994), Experience with artificial neural network models for short-term load forecasting in electrical power systems: a proposed application of expert networks, *IEEE Trans. on Power Systems*, U.K.

Read, E.G., (1998), Transmission pricing in New Zealand, in *Power Systems Restructuring: Engineering and Economics*, by Ilic, M., Galiana, F., Fink, L. (eds.), Kluwer Academic Publishers, Dordrecht, 264 - 280.

Reed, R.D., Marks, R.J., (1999), *Neural smithing: supervised learning in feedforward artificial neural networks*, The MIT Press, Cambridge, MA.

Ripley, B.D., (1996) *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.

Robinson, A.J., Fallside, F., (1991), A recurrent error propagation speech recognition system, *Computer Speech and Language*, 5, 259 - 274.

Robinson, D.A., (1992), Signal processing by neural networks in the control of eye movements, *Computational Neuroscience Symposium*, Indiana University-Purdue University at Indianapolis, 73 - 78.

Rogers, G.F.C., Mayhew, Y.R., (1967), *Engineering thermodynamics: work and heat transfer*, Longmans, London.

Rosenblatt, F., (1958), The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*, 65, 386 - 408.

Rosin, C., Belew, R., (1996), A competitive approach to game learning, *Proc. of the Ninth Annual ACM Conf. on Computational Learning Theory.*

Rumelhart, D., McClelland, J., (1986), *Parallel distributed processing*, MIT Press, Cambridge, MA.

Russel, N.T., Bakker, H.H.C., Chaplin, R.I., (2000), Modular neural networks modelling for long range prediction of an evaporator, *IEEE Trans. on Neural Networks*, 1.

Sakoe, H., Chiba, S., (1987), Dynamic programming algorithm optimization for spoken word recognition, *IEEE Trans. on Acoustics, Speech and Signal Processing*, 26, 43 - 49.

Sakr, M.F., Levitan, S.P., Chiarulli, D.M., Horne, B.G., Giles, C.L., (1997), Predicting multi-processor memory access patterns with learning models, *Proc. of the Fourteenth Int. Conf. on Machine Learning*, D.Fisher (eds.), Morgan Kaufmann, 305 - 312.

Samuel, A.L., (1959), Some studies in machine learning using the game of checkers, *IBM journal on Research and Development*, 3, 210 - 299. Reprinted in *Computers and Thought*, E.A.Feigenbaum and J.Feldman (eds), McGraw-Hill, New York

Sarle, W.S., (1994), Neural Networks and Statistical Models, in SAS Institute Inc., *Proc. of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC, 1538 - 1550, (also website ftp://ftp.sas.com/pub/neural/neural1.ps).

Sarle, W.S., (1997), *Bad science writing*, SAS Institute Inc., Cary, NC, USA.

Sarle, W.S., (1998), *FAQ artificial neural networks*, ftp://ftp.sas.com/pub/neural/ FAQ.html.

Sarle, W.S., (1999), *Ill-conditioning in neural networks*, SAS Institute Inc., Cary, NC, USA.

Saunders, G. M., Angeline, P. J., Pollack, J. B., (1996), Structural and behavioural evolution of recurrent networks, *Technical report, Laboratory for Artificial Intelligence*, Research Department of Computer and Information Science, Ohio State University.

Schoukens, J., Pintelton, R., (1991), *Identification of linear systems: A practical guideline to accurate modelling*, Pergamon, London.

Schraudolph, N.N., (1998), Centering neural network gradient factors, in *Neural Networks: tricks of the trade*, Orr and Mueller (eds), 207 – 226.

Schweppe, F.C., Caramanis, M.C., Tabors, R.D., Bohn, R.E., (1988), *Spot pricing of electricity*, Kluwer Academic Publishers, Boston.

Setiono, R., (1997), A penalty-function approach for pruning feedforward neural networks, *Neural Computation* 9, 185 - 204.

Shepherd, G.M., and Koch, C., (1990), *The Synaptic Organisation of the Brain*, New York: Oxford University Press.

Shiskin, J., Young, A.H., Musgrave, J.C., (1967), The X-11 variant of census method 11 seasonal adjustment program, *technical paper* 15, Washington DC, U.S. bureau of Census.

Siegelmann, H.T., Sontag, E.D., (1995), On the computational power of neural nets, *Journal of Computer and System Sciences*, Vol.50, No.1, 132 - 150.

Sietsma, J., Dow, R.F., (1988), Neural net pruning- why and how?, *Proceedings of the IEEE Int. Conf. on Neural Networks*, Vol.1, San Diego.

Simmons, J.A., and Saillant, P.A., (1992), Auditory deconvolution in echo processing by bats, *Computational Neuroscience Symposium*, Indiana University-Purdue University at Indianapolis, 15 - 32.

Simpson, P. K., (1990), *Artifical Neural Systems*, Pergamon Press.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B.,Glorrenec, P., Hjalmarsson, H., Juditsky, A., (1995), Non-linear blackbox modeling in system identification: A unified overview, *Automatica*, 31.

Sliwinski, B.J., Mech, A.R., Shih, R.S., (1978), Stratification in thermal storage during charging, *International heat transfer conference,* V4.

Söderström, T., Stoica, P., (1989), *System identification*, Prentice Hall Int., London.

Specht, D, (1992), Enhancements to probabilistic neural networks, *Proc. Int. Joint Conf. on Neural Networks*, Baltimore, MD.

Standards Association of New Zealand, (1988), *Low-pressure copper thermal-storage electric water heaters*, Wellington, New Zealand, NZ Standard 4602.

Stauffer, H.B., (1991), Smart enabling system for home automation, *IEEE Trans. on Consumer Electronics*, Vol.37, No.2, xxix – xxxv.

Stoecklein, A, Polard, A., Isaacs, N., Bishop, S., James, B., Ryan, G., Sanders, I., (1998), *Energy end-use and socio/demographic occupant characteristics of New Zealand households*, http://www.ema.org.nz/papers/98eesd.htm.

Ström, N., (1997a), Sparse connection and pruning in large dynamic artificial neural networks, *Proc. of Eurospeech '97*, 2807 - 2810.

Ström, N., (1997b), Phoneme probability estimation with dynamic sparsely connected artificial neural networks, *The Free Speech Journal*, Vol. 1, No.5.

Sullivan, J.B., (1996), *An Overview of DSM and Energy Services Activity in U.S. Utilities*, DA/DSM Europe 96.

Sutherland, J.L., (1991), *Stratification in a domestic hot water cylinder*, Final Year Project, Dept. of Mechanical Engineering, University of Canterbury, Christchurch, New Zealand.

Sutton, R.S., (1988), Learning to predict by methods of temporal differences, *Machine Learning*, Vol.3, 9 - 44.

Sutton, R.S., (1995), TD models: Modeling the world at a mixture of time scales, In *Proc. of the Twelfth Int. Conf. on Machine Learning*, Morgan Kaufmann.

Sutton, R.S., (1996), Generalization in reinforcement learning: Successful examples using sparse coarse coding, In *Advances in Neural Information Processing* 8. MIT Press.

Talukdar, S., Gellings, C. (1987), *Load Management,* IEEE Press.

Tan, H., Prokhorov, D.V., Wunsch II, D.C., (1995), Probabilistic and time-delay neural network techniques for conservative short-term stock trend prediction, *World Congress on Neural Networks'95*.

Tank, D.W., Hopfield, J.J., (1987), Collective computation in neuron-like circuits, *Scientific American*, December, 62 - 70.

Tank, D.W., Hopfield, J.J., (1987), Neural computation by concentrating information in time, *Proc. of the Int. Academy of Sciences*, USA, 84, 1896 - 1900.

Tepedelenlioglu, N., Rezgui, A., (1989), The effect of activation function on the back propagation algorithm, *Proceedings of IEEE International Conference on Systems Engineering.*

Tesauro, G. (1992), Practical issues in temporal difference learning, *Machine Learning* 8, Kluwer Academic Publishers, Boston, 257 - 277.

Tino, P., Horne, B.G., Giles, C.L., Collingwood, P.C., (1995), Finite state machines and recurrent neural networks- automata and dynamical systems approach, *technical report* UMAICS-TR-95-1, Institute for Advanced Computer Studies, University of Maryland, MD.

Tolat, V.V., Widrow, B., (1988), An adaptive broom balancer with visual inputs, *Proceedings of the IEEE International Conference on Neural Networks*, July, San Diego, CA, 641 - 647.

Tolley, D.L., (1987), The Basis for Load Management Terms in England and Wales, *IEE 5th International Conference on Metering Apparatus and Tariffs for Electricity Supply,* Edinburgh, 31 - 35.

Tromop, R, White, G., Gunn, C, (1995), Demand side management, *Proceedings of the IPENZ Annual Conference 1995,* Vol.1, Palmerston North, New Zealand.

Tsoi, A.C., Back, A.D., (1994), Locally recurrent globally feedforward networks: A critical review of architectures, *IEEE Transactions on Neural Networks*, Vol.5, No.2, 229 – 239.

Tveter, D., (1991), Getting a fast break with backprop, *AI Expert*, July.

Varfis, A., Versino, C., (1990), Univariate economic time series forecasting by connectionist methods, Int. Neural Network Conference, Paris, 342 – 345.

Waibel, A., (1989), Modular construction of time-delay neural networks for speech recognition, *Neural Computation*, 1, 39.

Walkington, M.T., (1990), *Forecasting electricity consumption with structural time series models*, unpublished Master of Science thesis, Victoria University, Wellington, New Zealand.

Wan, E.A., (1990), Temporal backpropagation for FIR neural networks, *IEEE International Joint Conference on Neural Networks*, Vol.1, San Diego, CA, 575 - 580.

Wan, E.A., (1994), Time series prediction by using a connectionist network with internal delay lines. In: A.S. Weigend, N.A. Gershenfeld, (eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, Reading, MA, 195 - 217.

Wang, P., Billinton, R., (2000), Optimum load-shedding technique to reduce the total customer interruption cost in a distribution system, *IEE Proceedings on Generation, Transmission and Distribution*, Vol.147, No.1, 51 - 56.

Weigend, A. S., Rumelhart, D. E., Huberman, B. A., (1991). Generalization by weight-elimination with application to forecasting. in R. P. Lippmann, J. Moody, D. S. Touretzky (eds.), *Advances in Neural Information Processing Systems* 3, Morgan Kaufmann, San Mateo, CA.

Weigend, A., (1994), On overfitting and the effective number of hidden units, *Proceedings of the 1993 Connectionist Models Summer School*, 335 - 342.

Welstead, S.T., (1994), *Neural network and fuzzy logic applications in C/C++*, Wiley.

WEMS, (1992), *Towards a competitive wholesale electricity market: conclusions and recommended approach*, Wholesale Electricity Market Study, final report, Wellington, New Zealand.

Werbos, P., (1974), *Beyond regression: new tools for prediction and analysis in the behavioural sciences*, Unpublished doctoral dissertation, Harvard University, MA.

Werbos, P., (1990), Backpropagation through time: what it does and how to do it, *Proceedings of the IEEE*, Vol.78, 1550 - 1560.

Westbury, S.E., (1994), *Competitive energy metering: a study of management issues*, unpublished Master of Engineering report, University of Canterbury, Christchurch, New Zealand.

Wettschereck, D., Dietterich, T., (1992), Improving the performance of radial basis function networks by learning centre locations, *Advances in Neural information processing systems* 4, J.E.Moody, S.J.Hanson, R.P.Lippman, (eds.) Morgan-Kaufmann, San Mateo, CA, 1133 - 1140.

Wezenberg, H., Dewe, M.B., (1995), Adaptive neural networks for tariff forecasting and energy management, *Int. Conference on Neural Networks*,Vol.2, Perth, Australia, 877 - 881.

White, D.A., Sofge, D.A., (eds.), (1992), *Handbook of intelligent control: neural, fuzzy, and adaptive approaches*, Van Nostrand- Reinhold, New York.

White, H., Gallant, A.R., (1992), On Learning the Derivatives of an Unknown Mapping with Multilayer Feedforward Networks, *Neural Networks* 5, 129 - 138.

William, R.J., Zipser, D., (1989), A learning algorithm for continually running fully recurrent neural networks, *Neural Computation* 1, 270 - 280.

Williams, R.J., Peng, J., (1990), An efficient gradient-based algorithm for on-line training of recurrent network trajectories, *Neural Computation*, Vol.2, 490 - 501.

Williams, R.J., Zipser, D., (1989), A learning algorithm for continually running fully recurrent neural networks, *Neural Computation*, Vol.1, 270 - 280.

Wilson, W.H., (1993), A comparison of architectural alternatives for recurrent networks, *Proceedings of the Fourth Australian Conference on Neural Networks*, ACNN'93, 189 -192.

Wilson, W.H., (1995), Stability of learning in classes of recurrent and feedforward networks, *Proceedings of the Sixth Australian Conference on Neural Networks*, ACNN'95, 142 - 145.

Winters, P.R., (1960), Forecasting sales by exponentially weighted moving averages, *Management Science*, 6.

Zaman, R., Wunsch, D.C, (1999), TD Methods Applied to Mixture of Experts for Learning 9x9 Go Evaluation Function, submitted for review to *1999 International Joint Conference on Neural Networks*, Washington DC. (also http://www.acil.ttu.edu/).

Zurigat, Y.H., Ghajar, A.J., Moretti, P.M., (1988), Stratified thermal storage tank inlet mixing characterization, *Applied Energy,* Vol.30, 99 – 111.

APPENDIX A : Thermistor specifications

Component specifications for the Philips negative temperature coefficient thermistor as used in the sensor strip installed on the hot water cylinder.

2322 642 6....

NTC THERMISTOR

disc

QUICK REFERENCE DATA

Resistance value at + 25 °C	3,3 Ω to 470 kΩ (E6 series)
$B_{25/85}$ value	2675 to 4650 K
Maximum dissipation	0,5 W
Dissipation factor	8,5 mW/K
Thermal time constant	≈ 17 s
Operating temperature range	
at zero power	−25 to + 125 °C
at maximum power	0 to + 55 °C

APPLICATION

Intended for general use.

DESCRIPTION

The thermistor has a negative temperature coefficient, it consists of a disc with two tinned copper wires. It is grey lacquered and colour coded, but not insulated.

MECHANICAL DATA

Outlines

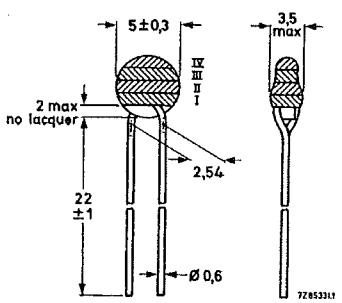


Fig. 1.

PACKAGING

500 thermistors in a cardboard box.

August 1983 181

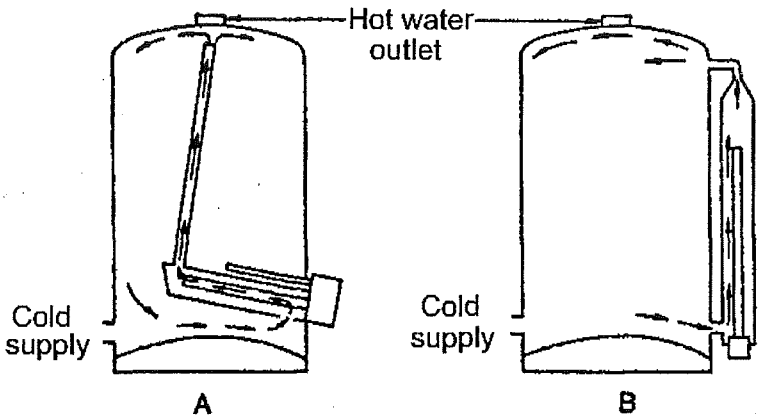
Table 1 Catalogue number 2322 642 6....

suffix of catalogue number	R_{25}	$B_{25/85}$ ± 5%	temperature coefficient	colour code (see Marking)		
	Ω	K	%/K	I	II	III
.102	1 000	3825	−4,3	brown	black	red

APPENDIX B : Quick Recovery Cylinders

Because in the standard hot water cylinder the heat from the element is spread through all the water in the cylinder, no hot water can be drawn off for some hours. In quick recovery units, however, the main heating element is fitted within an inner container of water that is separated from the surrounding water. As this smaller amount of water is heated to approximately 70°C it rises through a pipe directly to the top of the cylinder without dissipating heat to the surrounding water, and hot water accumulates at the top of the cylinder in amounts proportionate to the heating up time. Hot water can be drawn off in a relatively short time.

This type of heater is ideal where some hot water is wanted in a hurry, particularly in weekend baches, homes or cottages, where either all the hot Water has been used or the heater is only turned on at weekends or holiday times.



Two types of quick recovery cylinders.

This system is different from the booster units where extra elements are used to increase the heat-up time. Alternatively, some systems have 2 elements, one at high level and one at the bottom. This means hot water can be drawn off from the tap very quickly instead of having to wait for the whole cylinder of water to heat up.
(*source*: Plumbing and Gasfitting: volume 2 – Services and Roofing, by K. Doyle)

APPENDIX C :

A Project Proposal for a 3rd Pro. Student

Fluids Energy Management System (FEMS)

by H.Wezenberg, October 1998

Project summary

This project would suit a student keen to enhance his/her skills in 16 bit microcontroller development. The project aims to have a microprocessor based system build that is capable of accepting up to 20 analog input (AI) signals from temperature sensitive transistors/ICs. In addition a single AI signal comes from a relative humidity sensor. Existing AI hardware is available and should be catered for, but need not be seen as a limiting factor. The envisaged accompanying software must sample the inputs at a leisurely once a minute rate and do some signal processing to make the data obtained suitable for input to a, not to be included, artificial neural network (ANN). A digital output (DI) signal is required from the system to switch a 3 kW heating element. A year's worth of processed input data needs to be able to be stored in some form of memory (suggest EEPROM).
This project could eventually take on commercial significance and a cost effective (read cheap!) system should therefore be aimed for. It is intended that the developed system will become part of a Fluid Energy Management System (FEMS).

Background information

See Ass. Prof. Mike Dewe, papers are also available via him.

Available materials

Motorola HC16 microcontroller development kit.
A flexible strip fitted with 19 temperature sensors.
A humidity sensor

System requirements

Any microprocessor system designed should aim to incorporate the following features:

- Able to be connected up to 20 individual temperature sensors.
- Possess external circuitry needed for supply of power/current to the sensors.
- Process the temperature sensors signals to determine and subsequently store the actual temperatures measured by the sensors.
- Measure ambient relative humidity (RH) by means of the supplied Philips sensor (external circuitry needed).
- Process the RH sensor signal to obtain and subsequently store the actual rel. humid. measured by the sensor.

- Have a digital output capable of switching on/off a 3 kW heating element (external circuitry needed).
- Able to sample the temperatures and humidity on a once per minute basis.
- Every 30 minutes the sampled inputs are to be stored (on the hour and half hour).
- Stored data needs to be time-stamped (time and date).
- In view of the above requirements a real-time system clock is needed (suggest build as add-on board).
- Have enough memory (EEPROM) to store one year's worth of data.
- Be able to keep on operating if a domestic power supply failure occurs.
- Have some form of error verification/correction method to ensure valid the data storage.

System demonstration

The student should further demonstrate the viability of the system by writing a simple simulation software program in ANSI C (appropriately compiled by a suitable compiler) which switches the 3kw element ON in the night rate tariff time period 23:00 to 07:00 on Monday, Wednesday and Friday; but only when the average of the 19 temperatures sampled (once a minute) drops below, say, 40°C.

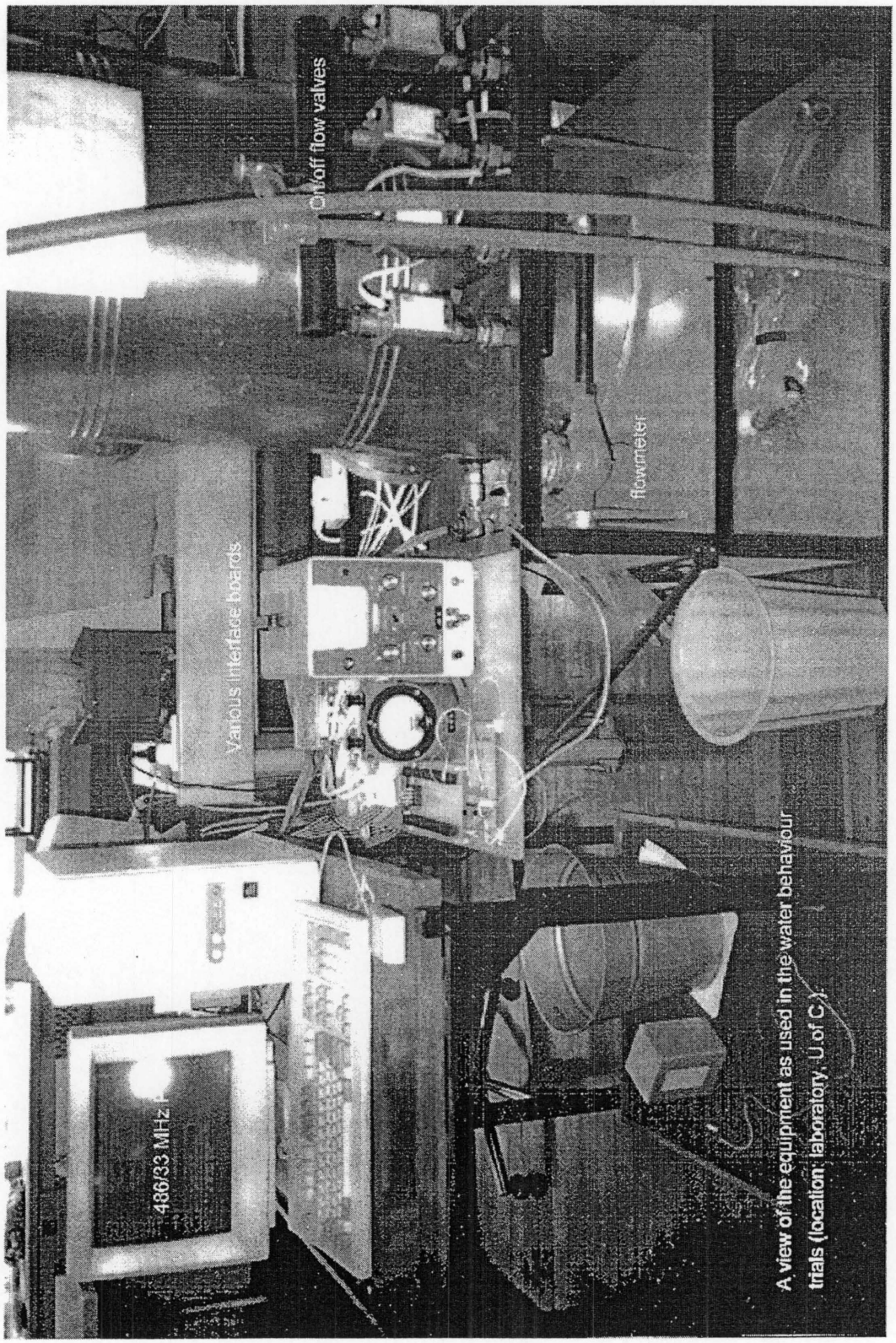
Further features

If time allows the system could be enhanced by the addition of a small LCD display which shows the date, time, temperatures and relative humidity measured (see similar display by Stephen Hunt).

Points to note

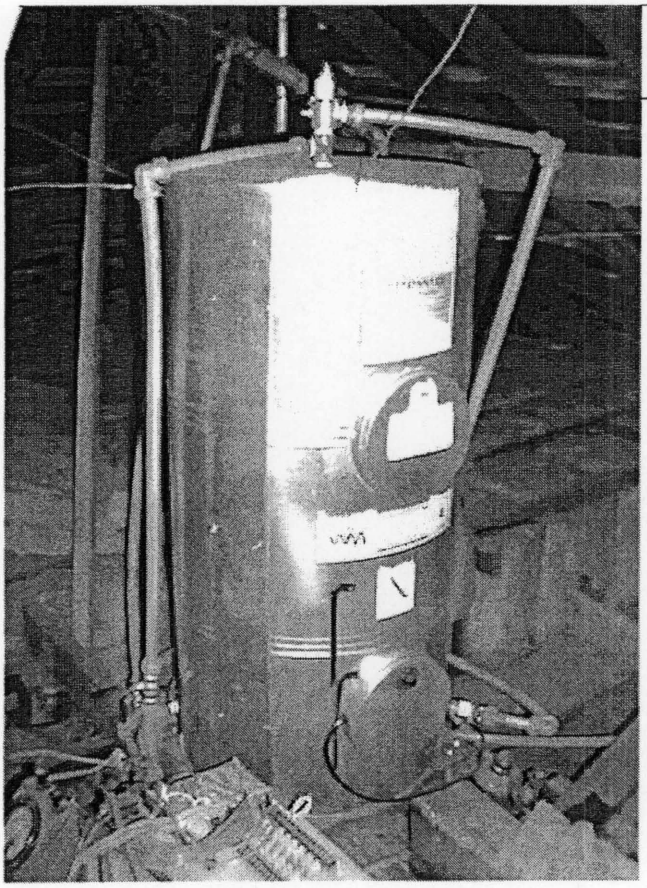
- It is envisaged that 19 of the 20 temperature inputs will measure temperatures in the range 10 to 95 °C. The remaining sensor will measure ambient (internal house) temperature, say, 0 to 30 °C. The relative humdity sensor can be anything from 10 to 100% (quoting from memory here! Check specs).
- Although NOT part of this project it is intended to use the 19 sensor temp's.(10 to 95 °C) to calculate the energy content of a domestic 180 litres hot water cylinder. The energy values can range from 0 to 60,000 kJ. The microprocessor then needs to convert any value in this range to an equivalent value between 0 and 1. These reduced values serve as an input to a Neural Network (ANN). For the ambient temp. the 0 to 30 °C will also be converted to an equivalent 0 to 1 range value for subsequent ANN usage. The same goes for the R.H. sensor. Given that these ANN input values need to be accurate to at least 3 decimal digits, can we possibly find a way to avoid having to use a floating point processing unit? One suggestion could be to make use of available floating point libraries (in C).

APPENDIX D : Test Environment (Lab)

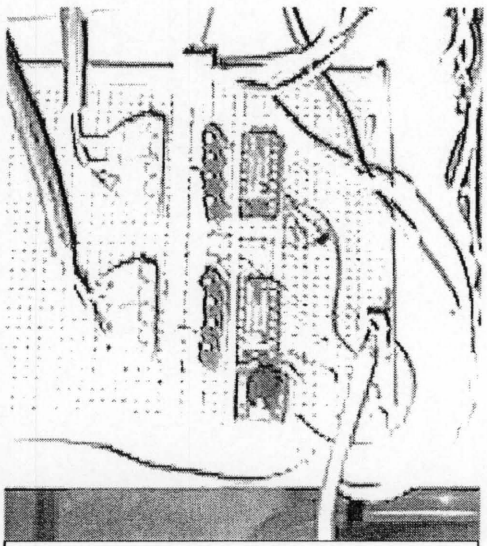


A view of the equipment as used in the water behaviour trials (location: laboratory, U.of C.).

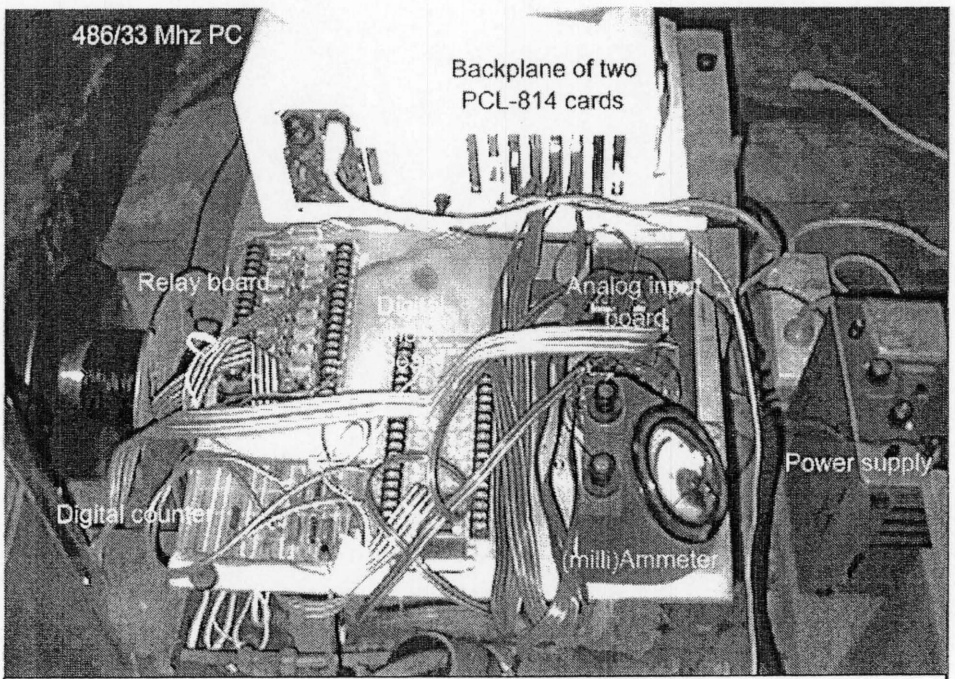
APPENDIX E : Test Environment (Attic)



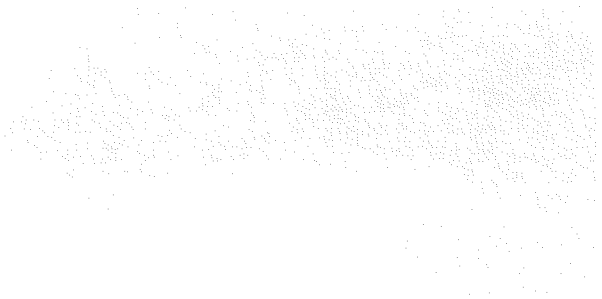
A view of the experimental hot water cylinder as installed in the attic.



Close-up view of the digital counter.

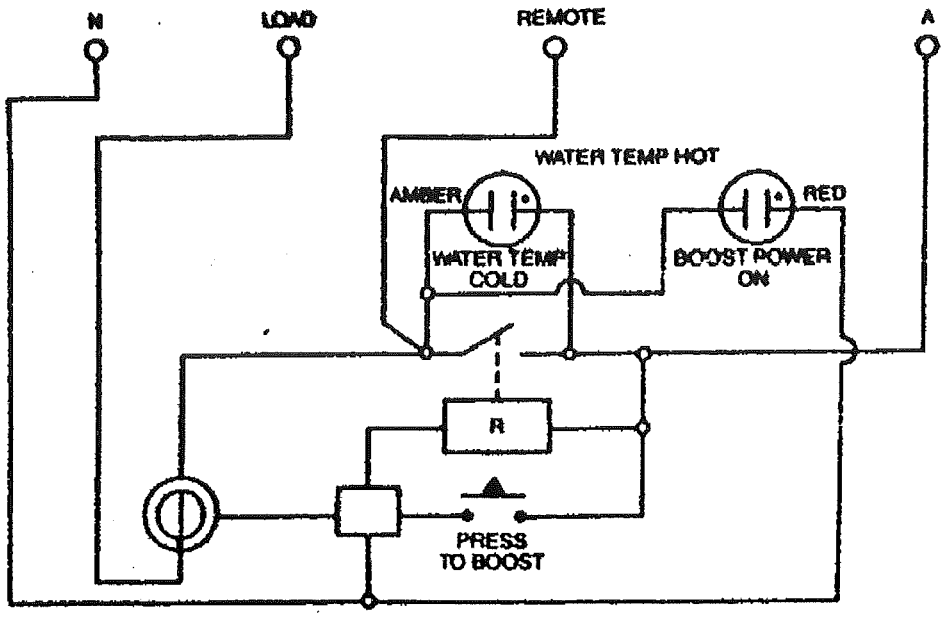


Overall view of the various boards used in the data acquisition phase.

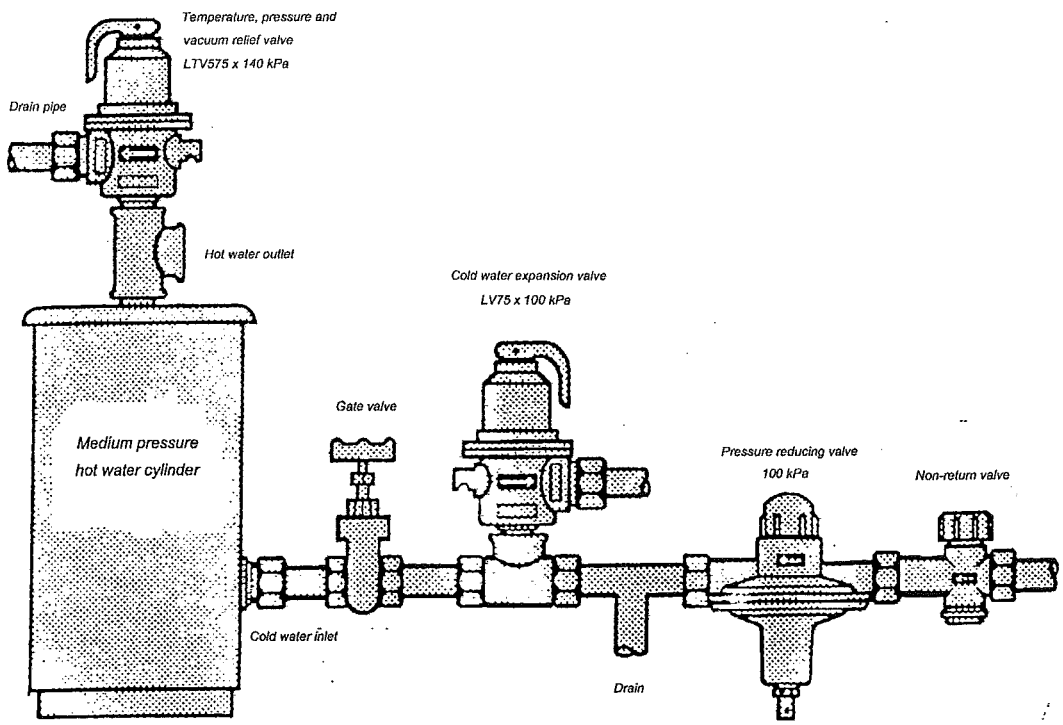


Two different views of the experimental hot water cylinder and the test equipment as installed in the attic.

APPENDIX F : Cylinder connection diagrams



Internal block diagram.



Installation set-up for a medium pressure hot water cylinder.

APPENDIX G :

Cylinder manufacturing specifications

SPECIFICATIONS – MEDIUM PRESSURE (12m head) ELECTRIC WATER HEATER NZS 4602

Cylinder Manufacturer	Multi Machinery Limited, Christchurch, New Zealand
Copper Barrel	
Cylinder Volume	180 litres
Diameter	460 mm
Length	965 mm
Copper Gauge	0.9 mm
Overlap	5 mm x 2
Joint	silver brazing alloy
Dome	Diameter 460 mm Depth 110 mm Thickness 1.2 mm
Insulation	50 mm wall thickness, formed by mixing Daltolac GP 17 and Suprasec 50005 ratio 1:1 at 20°C
Case	Diameter 560 mm Ends 562 mm Height - Length 1200 mm Colour steel sheet 0.55 mm
General	Bottom Inlet 20 mm BSP Female with internal baffle approx. 100 mm sq. Outlet 20 mm BSP Female Thermostat Pocket - 290 mm Length 10 mm OD Element Flange 32 mm BSP Earth Strap 100 mm x 15 mm x .7 copper

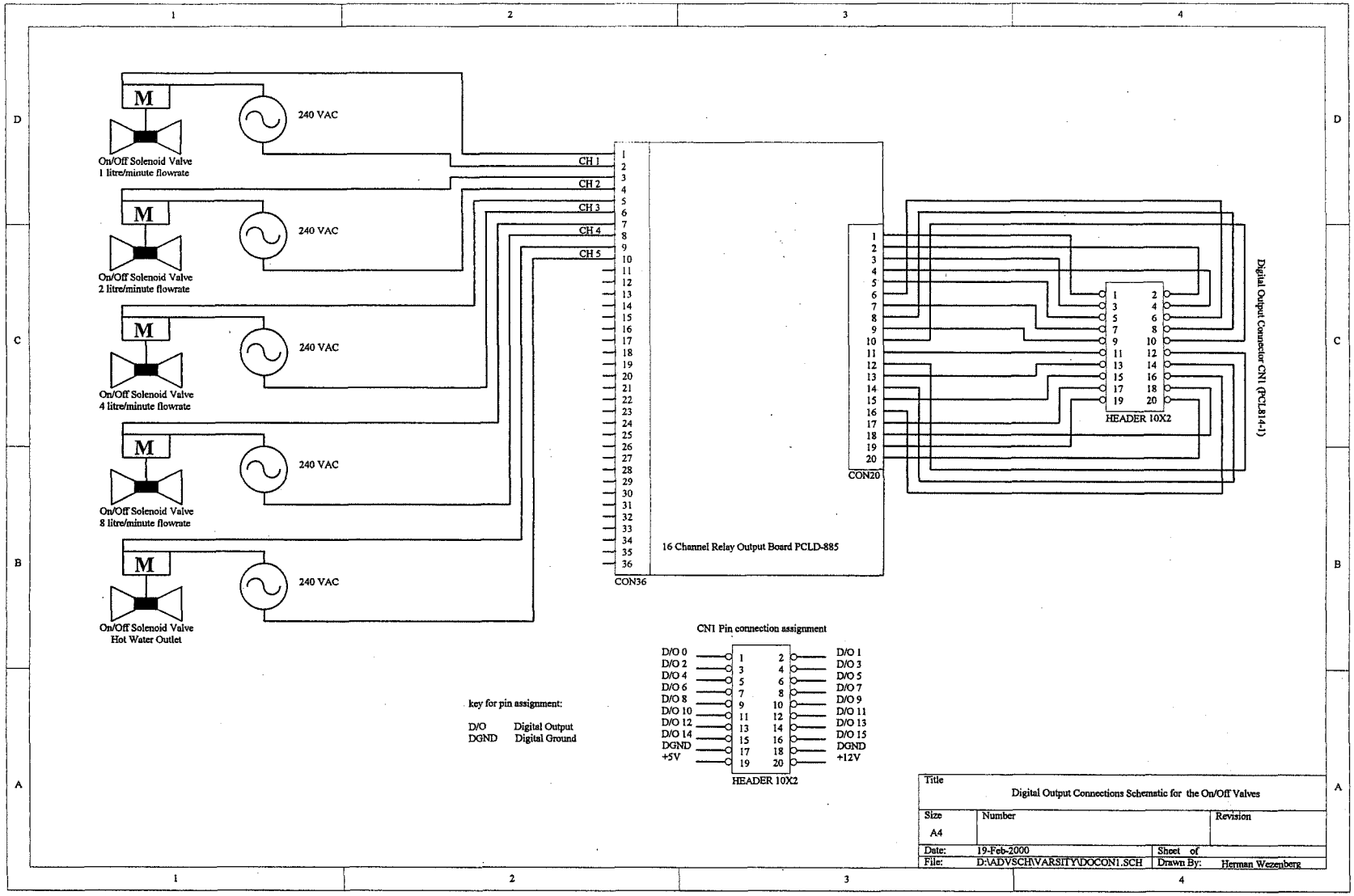
Specifications of Materials

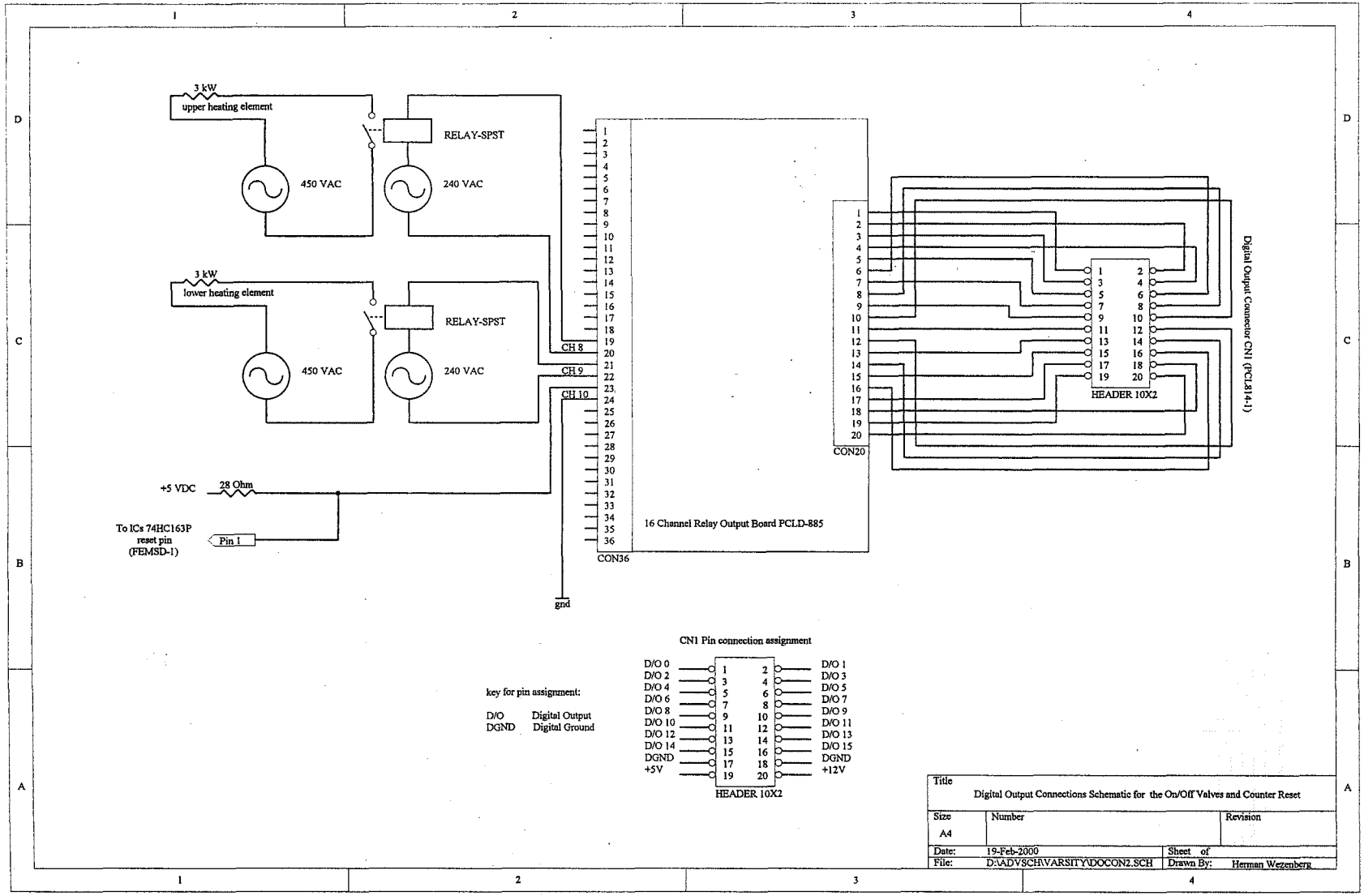
Copper to be purchased from North Trading Marketing Ltd, Christchurch

	Copper Alloy 102 (half hard) England – Italy J I S H3100-C-1020 P - Japan
Mico Wakefield Limited	
	England - Italy (half hard) No 101 99.9% copper
Fittings	J& T Christie Ltd (Giltech Precision Castings Ltd Dunedin)
	Simsmetal Industries Ltd - Spartan Engineering
Brazing Alloys	Huthnance-Jamac (Englehard)-N.Z.I.G. (Silfos 5) Dominion Lead (AS 1167.1 1984 / B51845.84 CPI)

APPENDIX H : Detail schematics and circuit diagrams

- Digital Output Connections Schematic for the On/Off valves.
- Digital Output Connections Schematic for the On/Off valves and Counter Reset.
- CN3 Pin Connection Assignment (Analog Input Plug).
- Analog Input Connections For The Hot Water Cylinder Thermistors.
- Digital Input Connections Schematic For The Cold Water Inlet Flow Meter.
- Circuit Diagram For The Constant Current Source Board FEMSD-1.
- Circuit Diagram For The 8-Bit Binary Flow Counter Board FEMSD-2.

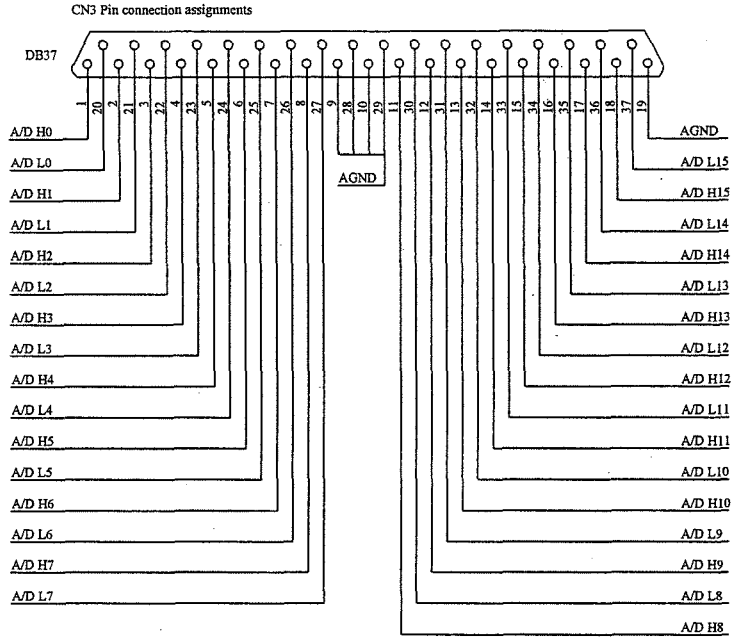

M
240 VAC
On/Off Solenoid Valve
1 litre/minute flowrate
On/Off Solenoid Valve
2 litre/minute flowrate
On/Off Solenoid Valve
4 litre/minute flowrate
On/Off Solenoid Valve
8 litre/minute flowrate
On/Off Solenoid Valve
Hot Water Outlet
CH 1
CH 2
CH 3
CH 4
CH 5
CON36
16 Channel Relay Output Board PCLD-885
CON20
HEADER 10X2
Digital Output Connector CN1 (PCL814+1)
CN1 Pin connection assignment
D/O 0
D/O 1
D/O 2
D/O 3
D/O 4
D/O 5
D/O 6
D/O 7
D/O 8
D/O 9
D/O 10
D/O 11
D/O 12
D/O 13
D/O 14
D/O 15
DGND
DGND
+5V
+12V
HEADER 10X2
key for pin assignment:
D/O Digital Output
DGND Digital Ground
Title
Digital Output Connections Schematic for the On/Off Valves
Size
A4
Number
Revision
Date: 19-Feb-2000
Sheet of
File: D:\ADVSCH\VARSITY\DOCON1.SCH
Drawn By: Herman Wezenberg


3 kW
upper heating element
450 VAC
RELAY-SPST
240 VAC
3 kW
lower heating element
450 VAC
RELAY-SPST
240 VAC
CH 8
CH 9
CH 10
+5 VDC
28 Ohm
To ICs 74HC163P
reset pin
(FEMSD-1)
Pin 1
gnd
16 Channel Relay Output Board PCLD-885
CON36
CON20
HEADER 10X2
Digital Output Connector CN1 (PCL814-1)
CN1 Pin connection assignment
D/O 0
D/O 2
D/O 4
D/O 6
D/O 8
D/O 10
D/O 12
D/O 14
DGND
+5V
D/O 1
D/O 3
D/O 5
D/O 7
D/O 9
D/O 11
D/O 13
D/O 15
DGND
+12V
HEADER 10X2
key for pin assignment:
D/O Digital Output
DGND Digital Ground
Title
Digital Output Connections Schematic for the On/Off Valves and Counter Reset
Size
A4
Number
Revision
Date: 19-Feb-2000
Sheet of
File: D:\ADVSCH\VARSITY\DOCON2.SCH
Drawn By: Herman Wezenberg

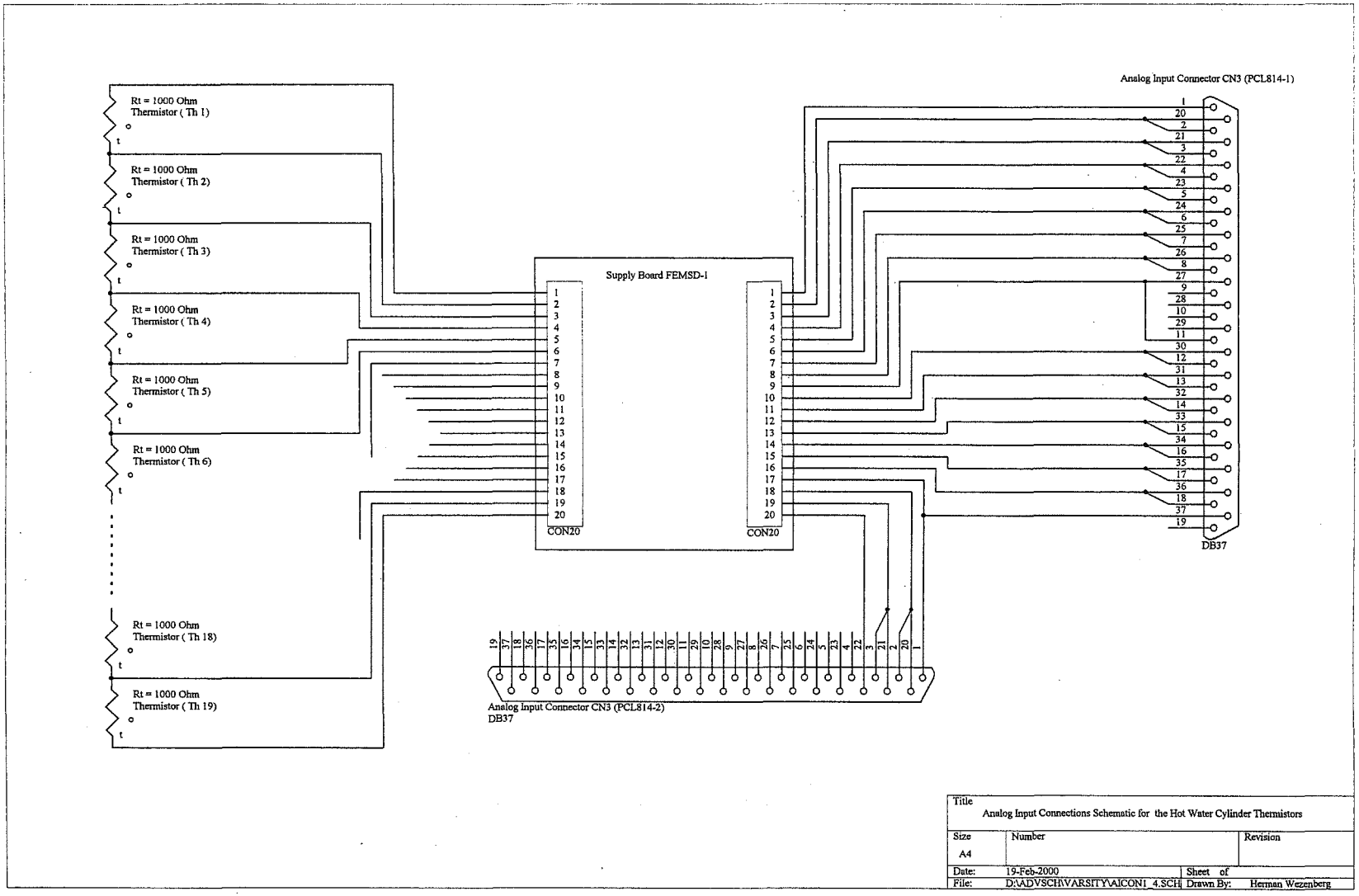
key for pin assignment:

A/D H Analog Input High (differential)
A/D L Analog Input Low (differential)
AGND Analog Ground

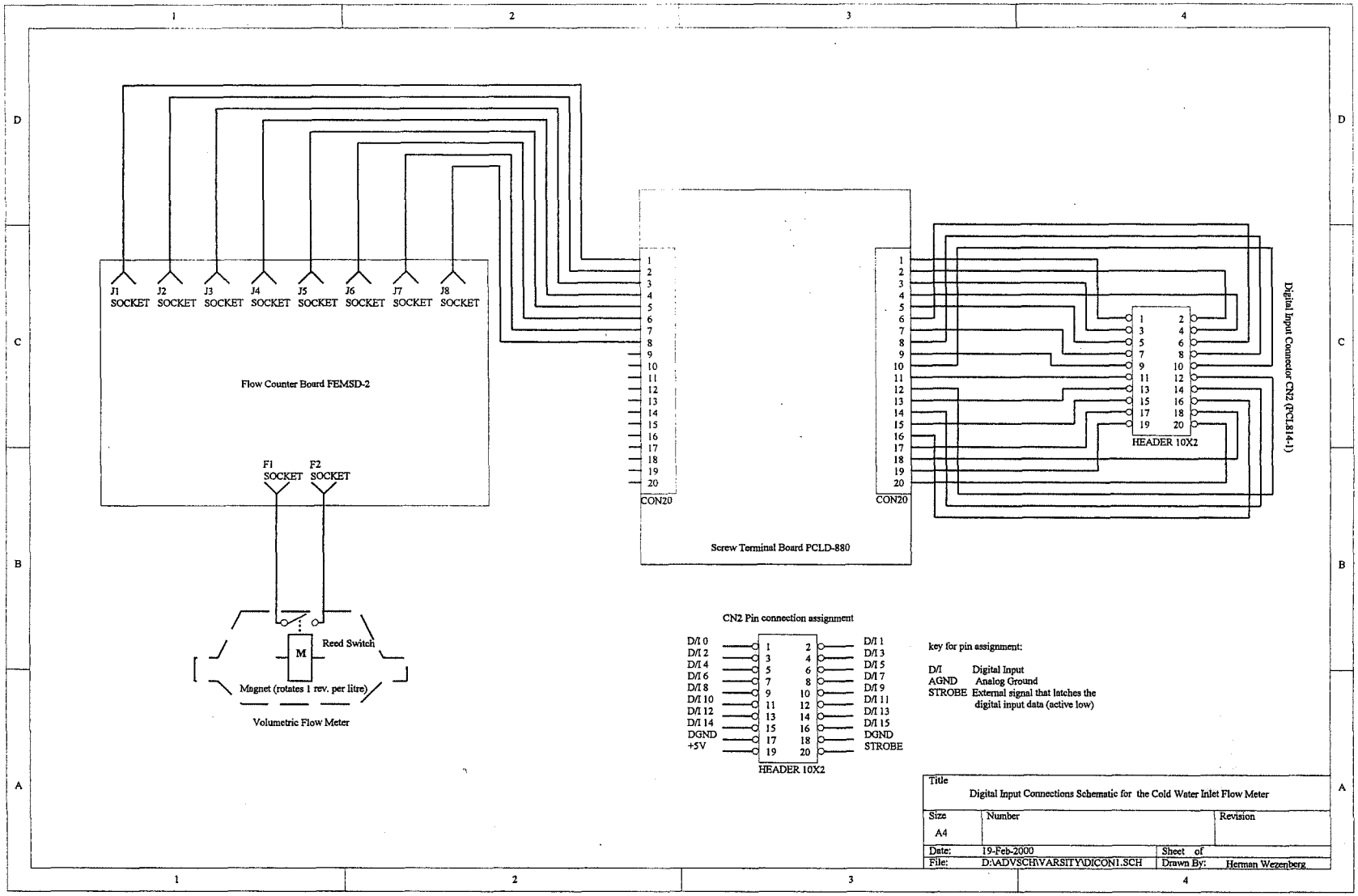
CN3 Pin connection assignments

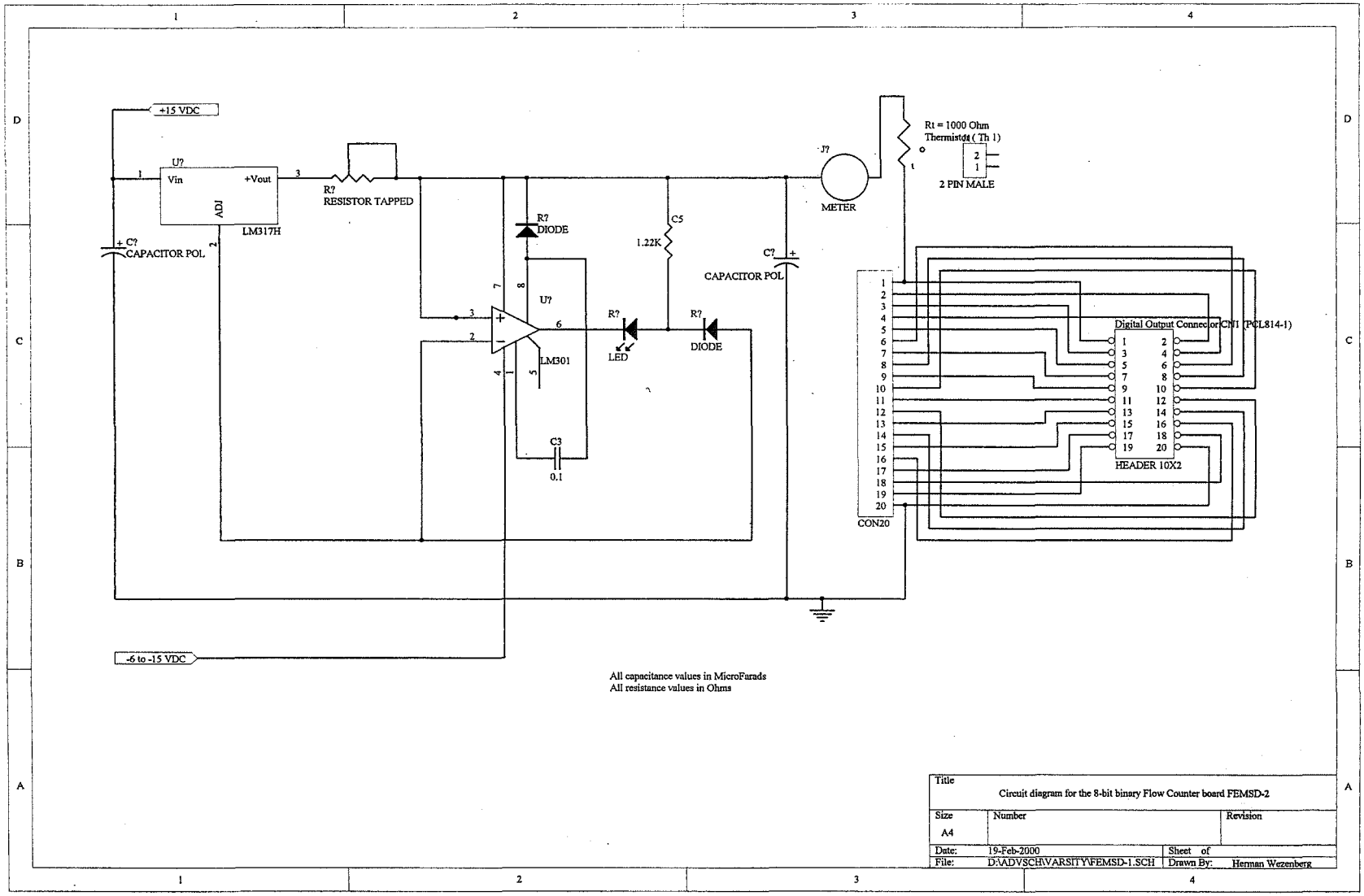


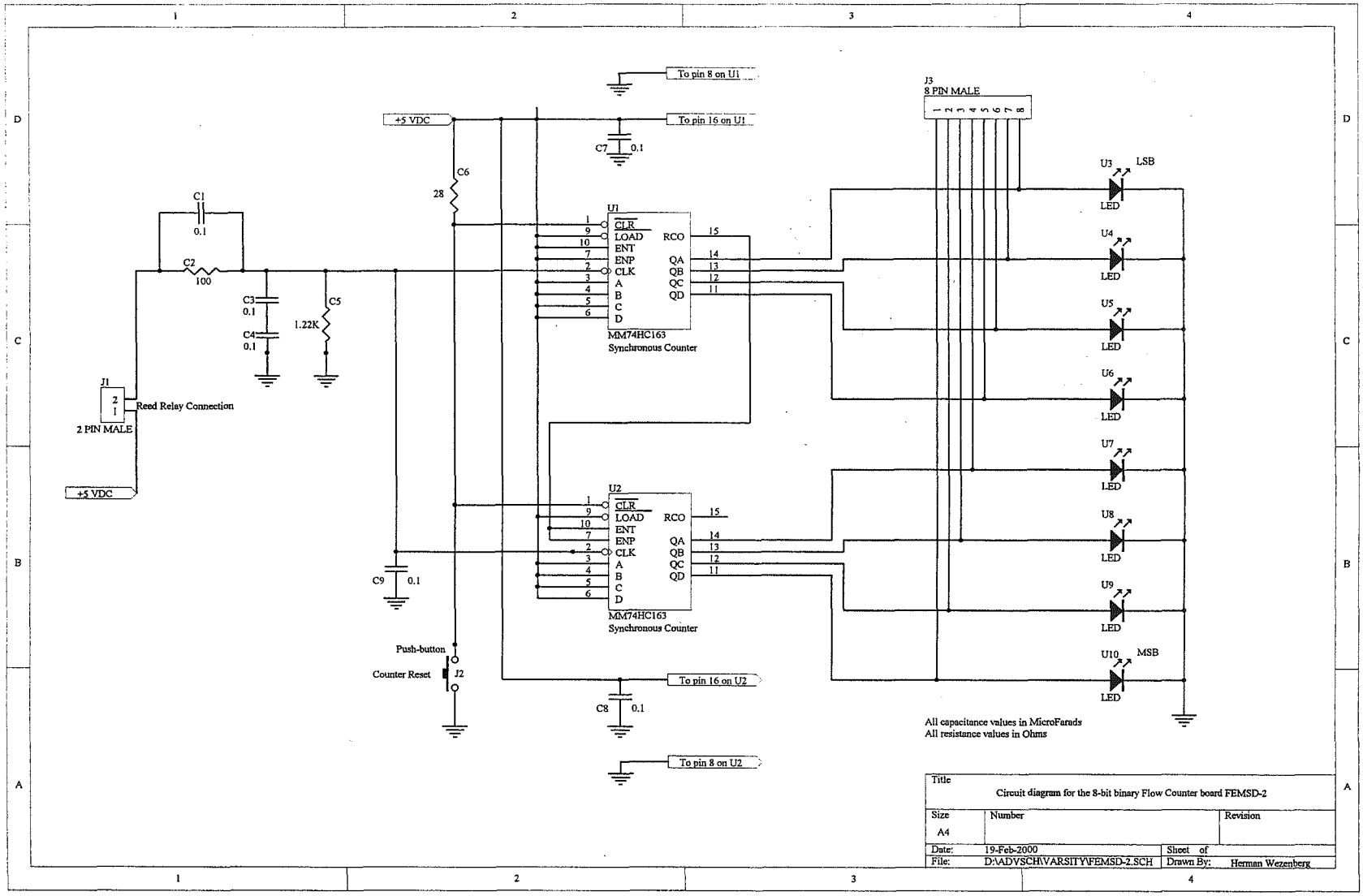
Title	CN3 Pin Connection Assignments (Analog Input plug)	
Size A4	Number	Revision
Date: 19-Feb-2000	Sheet of	
File: D:\ADVSCH\VARSITY\KEY_CON1.SCH	Drawn By: Herman Wezenberg	



Title	Analog Input Connections Schematic for the Hot Water Cylinder Thermistors		
Size A4	Number		Revision
Date:	19-Feb-2000	Sheet of	
File:	D:\ADVSCH\VARSITY\AICON1_4.SCH	Drawn By:	Herman Wezenberg


J1 SOCKET
J2 SOCKET
J3 SOCKET
J4 SOCKET
J5 SOCKET
J6 SOCKET
J7 SOCKET
J8 SOCKET
Flow Counter Board FEMSD-2
F1 SOCKET
F2 SOCKET
Reed Switch
M
Magnet (rotates 1 rev. per litre)
Volumetric Flow Meter
CON20
CON20
Screw Terminal Board PCLD-880
HEADER 10X2
Digital Input Connector CN2 (PCL814-1)
CN2 Pin connection assignment
D/I 0
D/I 2
D/I 4
D/I 6
D/I 8
D/I 10
D/I 12
D/I 14
DGND
+5V
D/I 1
D/I 3
D/I 5
D/I 7
D/I 9
D/I 11
D/I 13
D/I 15
DGND
STROBE
HEADER 10X2
key for pin assignment:
D/I Digital Input
AGND Analog Ground
STROBE External signal that latches the digital input data (active low)
Title
Digital Input Connections Schematic for the Cold Water Inlet Flow Meter
Size A4
Number
Revision
Date: 19-Feb-2000
Sheet of
File: D:\ADVSCH\VARSITY\DICON1.SCH
Drawn By: Herman Wezenberg


+15 VDC
U?
Vin
+Vout
ADJ
LM317H
R?
RESISTOR TAPPED
+ C?
CAPACITOR POL
R?
DIODE
C5
1.22K
C?
CAPACITOR POL
J?
METER
Rt = 1000 Ohm
Thermistor (Th 1)
2 PIN MALE
U?
LM301
R?
LED
R?
DIODE
C3
0.1
CON20
Digital Output Connector CN1 (PCL814-1)
HEADER 10X2
-6 to -15 VDC
All capacitance values in MicroFarads
All resistance values in Ohms
Title
Circuit diagram for the 8-bit binary Flow Counter board FEMSD-2
Size
A4
Number
Revision
Date: 19-Feb-2000
Sheet of
File: D:\ADVSCH\VARSITY\FEMSD-1.SCH
Drawn By: Herman Wezenberg


To pin 8 on U1
+5 VDC
To pin 16 on U1
C7 0.1
C6
28
C1
0.1
C2
100
C3
0.1
C4
0.1
C5
1.22K
U1
CLR
LOAD
ENT
ENP
CLK
A
B
C
D
RCO
QA
QB
QC
QD
MM74HC163
Synchronous Counter
J1
2
1
2 PIN MALE
Reed Relay Connection
+5 VDC
U2
MM74HC163
Synchronous Counter
C9 0.1
Push-button
Counter Reset
J2
To pin 16 on U2
C8 0.1
To pin 8 on U2
J3
8 PIN MALE
U3 LSB
U4
U5
U6
U7
U8
U9
U10 MSB
LED
All capacitance values in MicroFarads
All resistance values in Ohms
Title
Circuit diagram for the 8-bit binary Flow Counter board FEMSD-2
Size
A4
Number
Revision
Date: 19-Feb-2000
Sheet of
File: D:\ADVSCH\VARSITY\FEMSD-2.SCH
Drawn By: Herman Wezenberg

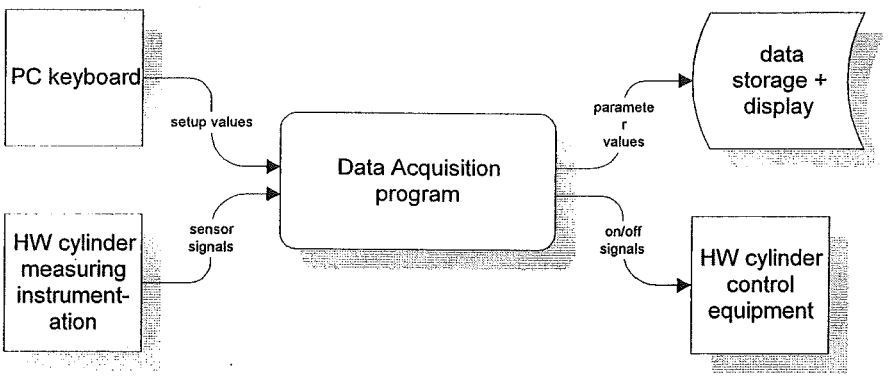
APPENDIX I :

Flow diagrams, activity specifications, and data dictionaries.

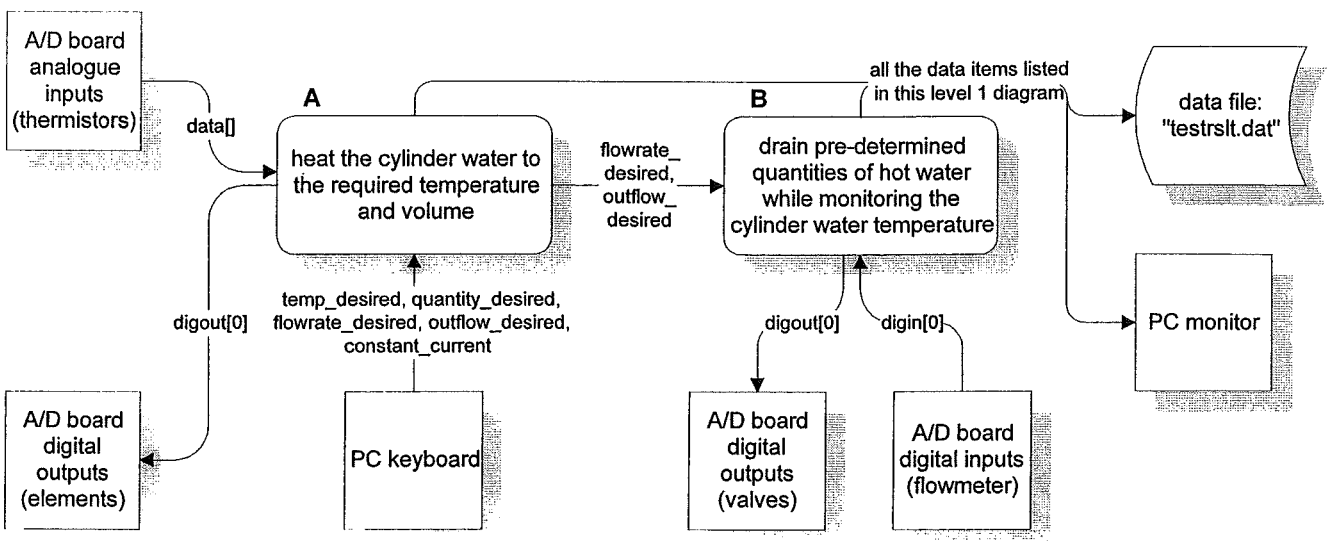
- Hot Water Cylinder Data Acquisition Program.
- Hot Water Cylinder Linear Prediction Program.
- Hot Water Cylinder Fluid Energy Management Program.

Hot Water Cylinder Data Acquisition Program

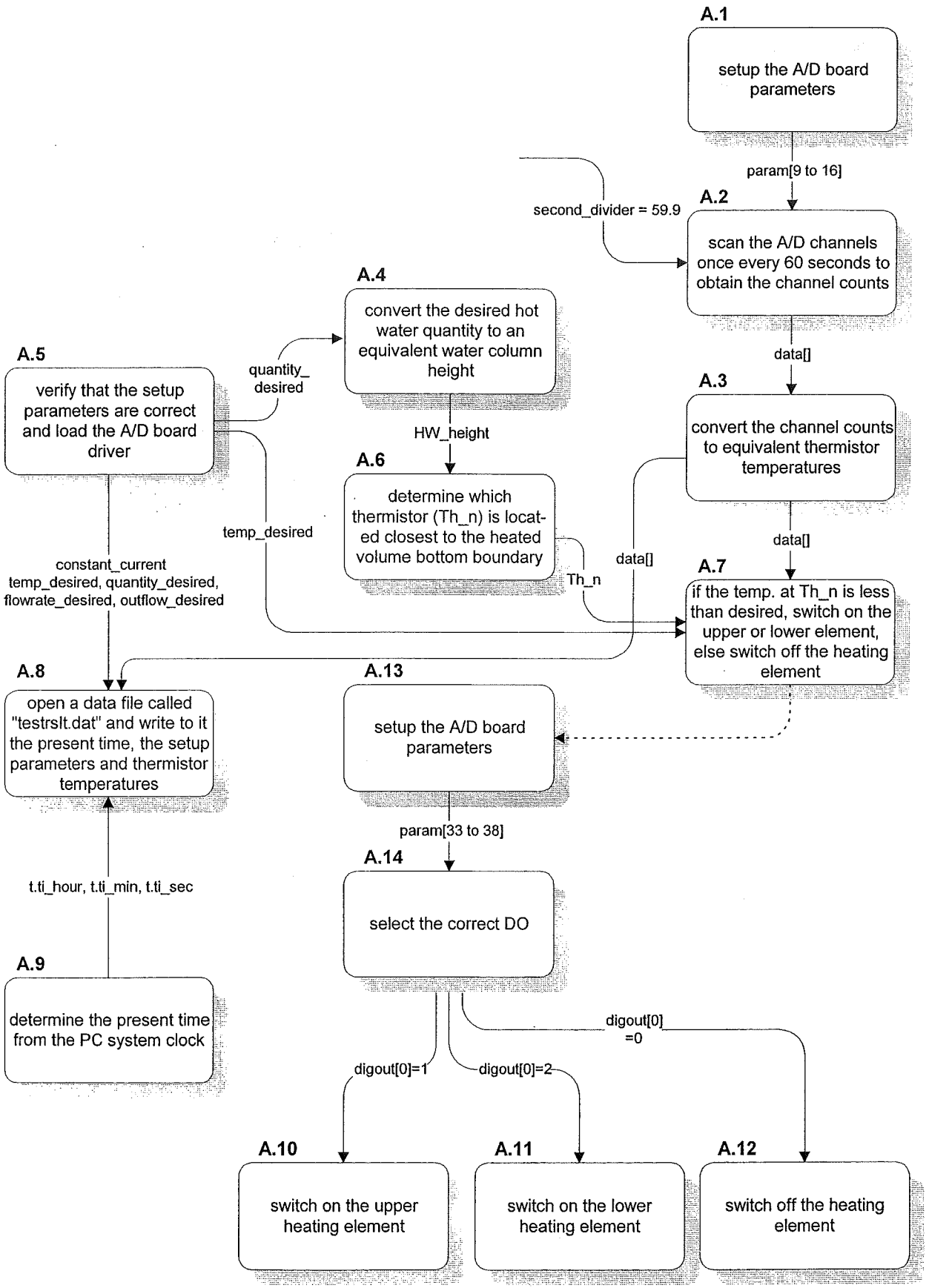
High Level Data Flow diagram



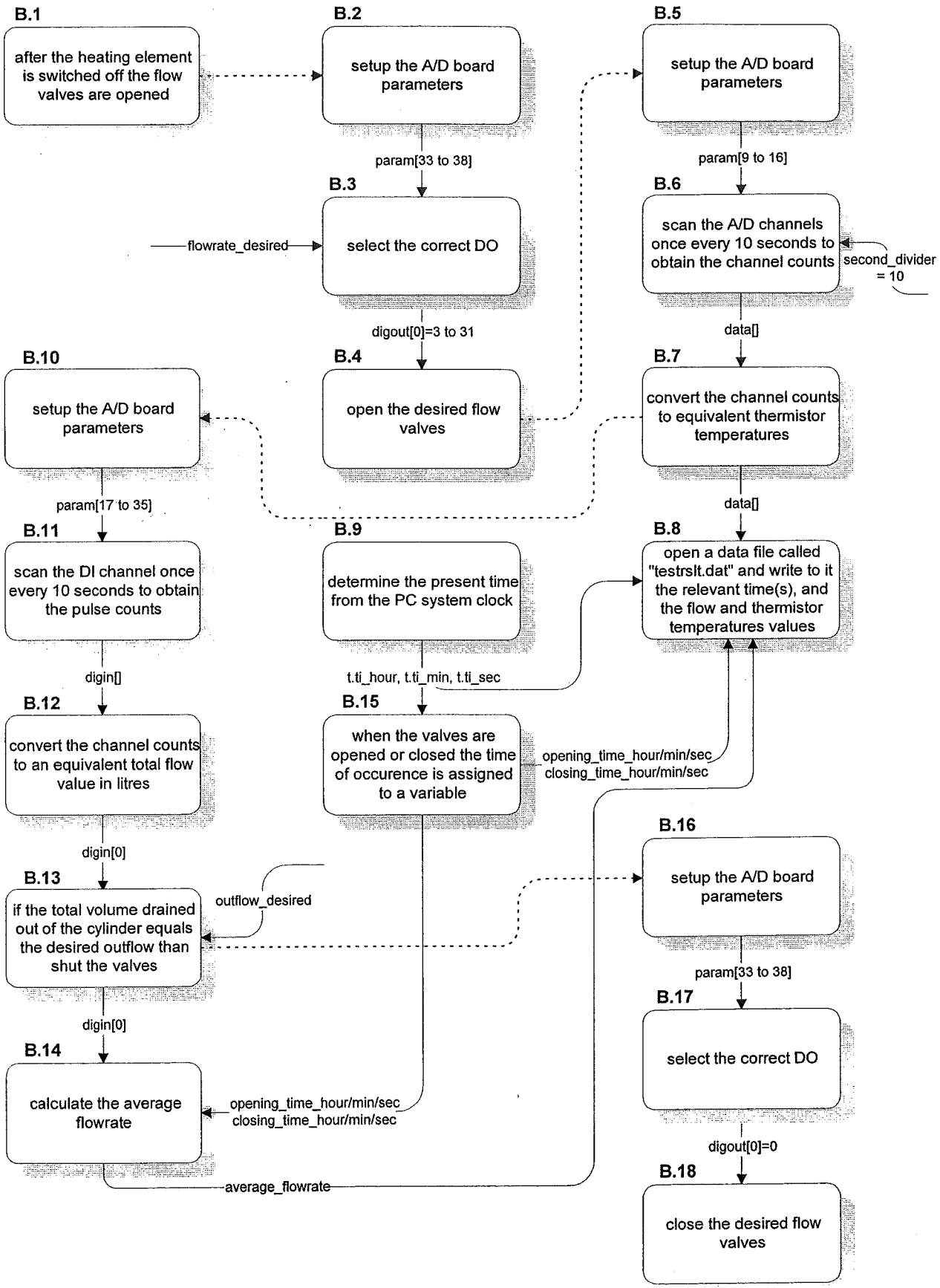
Level 1 Data Flow diagram

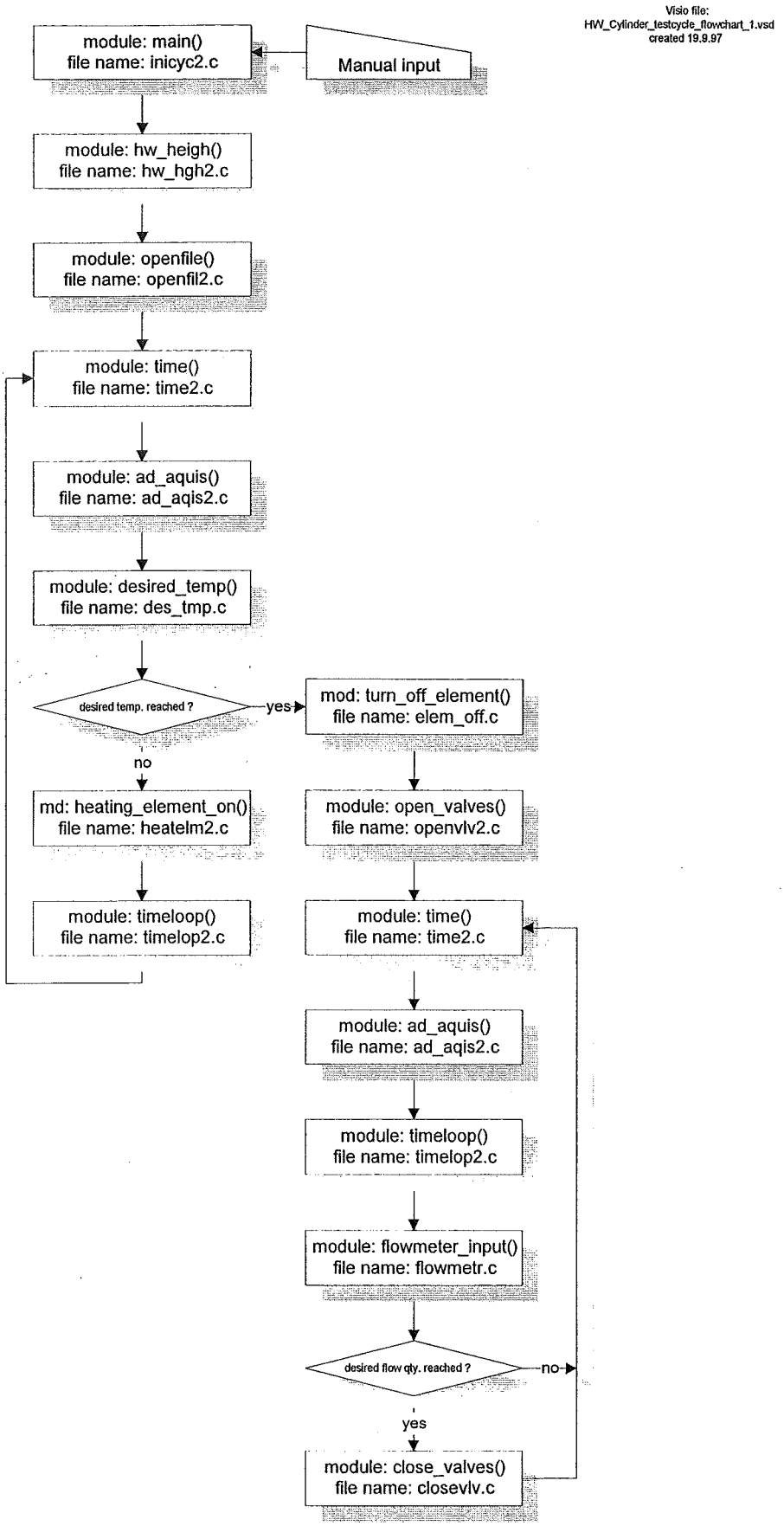


Level 2 Data Flow diagram for 'A'

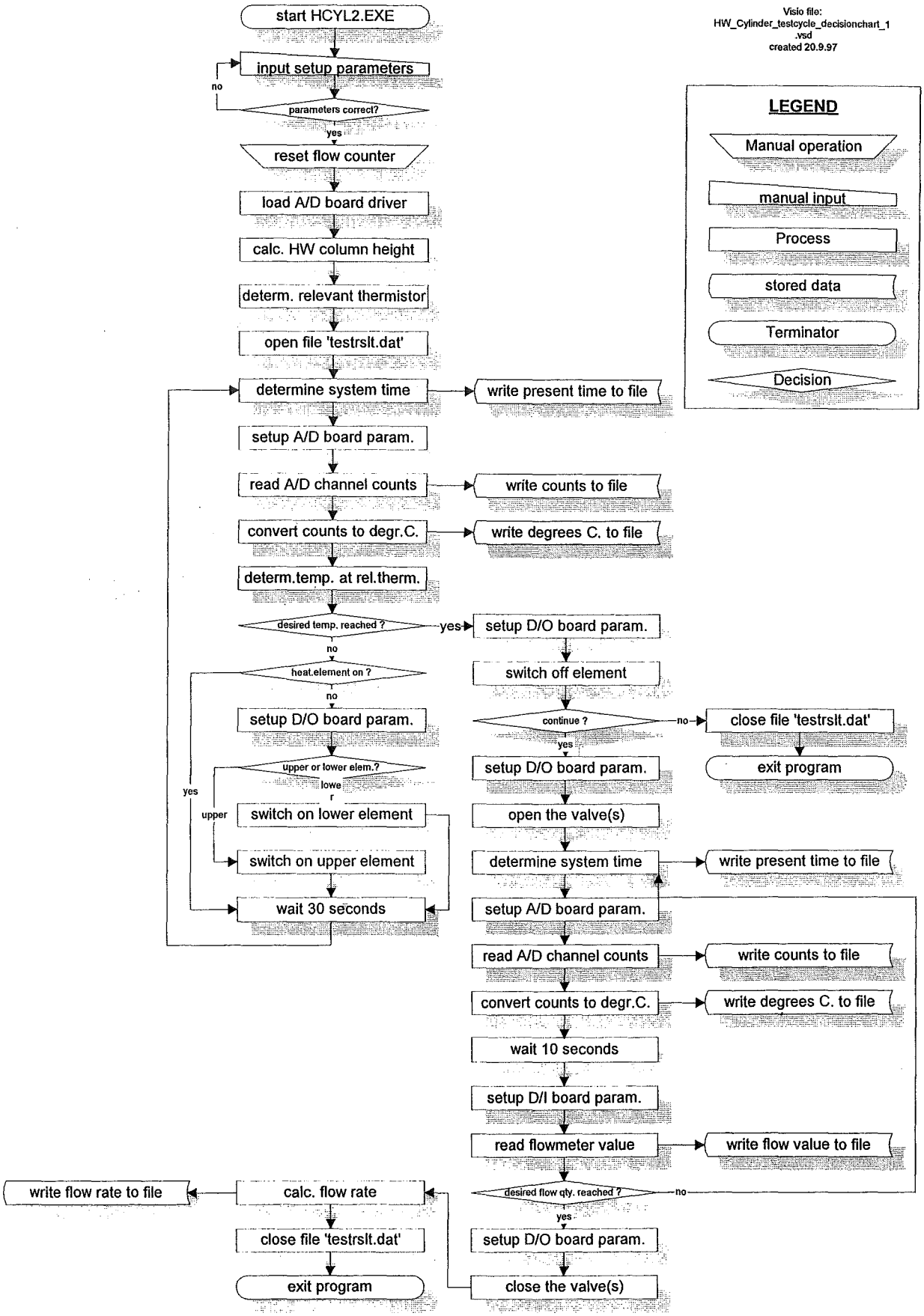


Level 2 Data Flow diagram for 'B'





A flow diagram for the 'C' modules as used in the software for testing water behaviour in a hot water cylinder.



A flow diagram that shows the activities and decisions undertaken by the software for testing water behaviour in a hot water cylinder.

Activity specification: description of the basic activities related to the HW cylinder data acquisition program.

A.1 Setup the A/D board parameters

module: ad_aquis()

description: Prior to inputting or outputting data the software needs to set up a number of parameters which tell the A/D board what task to perform and how to go about it. For analogue input the array elements 9 to 16 of param[] need to be specified.

A.2 Scan the A/D channels once every 60 seconds to obtain the channel counts

module: ad_aquis() and timeloop()

description: During the heating stage of the test cycle a thermistors temperature reading once a minute was deemed sufficient. The module timeloop() loops on itself while checking the PC system clock for the elapse of 59.9 seconds; it then allows ad_aquis() to scan the AI channels of the A/D board.

A.3 Convert the channel counts to equivalent thermistor temperatures

module: ad_aquis()

description: The following equation is used to convert the AI channel counts to the temperatures as sensed by the thermistors.

$$\text{Temp. (Celsius)} = \frac{3825}{\ln\left(\frac{\text{channel count}}{34.86\ I}\right)} - 273 \quad \text{, where I = constant\_current in Amps}$$

A.4 Convert the desired hot water quantity to an equivalent water column height

module: hw_heigh()

description: Both the cylinder diameter and its height are known. The volume of hot water needed is given in litres and the program needs to determine how far down the length of the cylinder this quantity of hot water will extend, starting from the top of the cylinder.

A.5 Verify that the setup parameters are correct and load the A/D board driver

module: main()

description: The user is asked to check and confirm the test setup parameters entered via the keyboard. If not correct, alterations to the values can be made. Subsequently the choice is given whether or not the program needs to load the software driver for the A/D data acquisition board.

A.6 Determine which thermistor (Th_n) is located closest to the heated volume bottom boundary

module: hw_heigh()

description: Having determined the length of the hot water column in activity A.4, the thermistor which is nearest to the bottom boundary of that column can be determined by dividing the column length by the distance separating each thermistor (5 cm). This thermistor is than designated as Th_n.

A.7 If the temp. at Th_n is less than desired, switch on the upper or lower element, else switch off the heating element

module: desired_temp()

description: The value of the water temperature measured by thermistor Th_n is compared with the desired temperature as set by the user. As the upper element is level with the position of thermistor no.7 the program will check whether thermistor Th_n is higher or lower relative to no.7; if higher than the upper element will be activated, else the lower element. If the temperature is too low a heating element will need to be switched on. When the water is already at, or has reached, the temperature set by the user, the heating element will need to be switched off. This command will be given regardless of whether an element has, or has not, been switched on in the first place.

A.8 Open a data file called "testrslt.dat" and write to it the present time, the setup parameters and thermistor temperatures

module: openfile()

description: A file is opened on the hard drive of the PC running the program. All the setup values and the first temperatures are written to it, together with a timestamp.

A.9 Determine the present time from the PC system clock

module: time()

description: The value of real time in hour:min:sec can be determined from the PC system clock by using the appropriate 'C' commands.

A.10 Switch on the upper heating element

module: heating_element_on()

description: The upper heating element is connected to a set of relays, the smaller of which is operated directly by the DO section of the A/D board. The value of digout[0] = 2 will switch the upper element on.

A.11 Switch on the lower heating element

module: heating_element_on()

description: The lower heating element is connected to a set of relays, the smaller of which is operated directly by the DO section of the A/D board. The value of digout[0] = 1 will switch the lower element on.

A.12 Switch off the heating element

module: turn_off_element()

description: Outputs the value of digout[0] = 0 to deactivate all relays of the appropriate DO section, thereby switching off the heating element.

A.13 Setup the A/D board parameters

module: heating_element_on() and turn_off_element()

description: Prior to inputting or outputting data the software needs to set up a number of parameters which tell the A/D board what task to perform and how to go about it. For digital output the array elements 33 to 38 of param[] need to be specified.

A.14 Select the correct DO

module: heating_element_on() and turn_off_element()

description: Which heating element is switched on or off is determined by the value given to the array element digout[0], (e.g. a value of 2 will switch the upper element on). The correct value needs to be specified for the switching action that is desired.

B.1 After the heating element is switched off the flow valves are opened

module: turn_off_element()

description: The program ensures that the heating element is switched off thereby completing the 'heating cycle' of the test program and proceeds to continue with the 'flow cycle' section.

B.2 Setup the A/D board parameters

module: open_valves()

description: Prior to inputting or outputting data the software needs to set up a number of parameters which tell the A/D board what task to perform and how to go about it. For digital output the array elements 33 to 38 of param[] need to be specified.

B.3 Select the correct DO

module: open_valves()

description: The value of digout[0] is set to a number which has been determined from a table of values. The specific value chosen will depend on the value for the desired flow rate (the setup parameter 'desired_flowrate'). For instance, digout[0] = 21 will open valves B and D which will result in a flow rate of 12 litres per minute.

B.4 Open the desired flow valves

module: open_valves()

description: Each flow valve is connected to a relay, which is operated directly by the DO section of the A/D board. The value given to digout[0] will turn open any number of the four valves available. The four valves are set to give fixed flows of 1, 2, 4, or 8 litres/min, thus allowing a combined flow rate of between 1 and 15 litres/min.

B.5 Setup the A/D board parameters

module: ad_aquis()

description: Prior to inputting or outputting data the software needs to set up a number of parameters which tell the A/D board what task to perform and how to go about it. For analogue input the array elements 9 to 16 of param[] need to be specified.

B.6 Scan the A/D channels once every 10 seconds to obtain the channel counts

module: ad_aquis() and timeloop()

description: During the flow stage of the test cycle a thermistor temperature reading once every 10 seconds was used. The module timeloop() loops on itself while checking the PC system clock for the elapse of 10.0 seconds; it then allows ad_aquis() to scan the AI channels of the A/D board.

B.7 Convert the channel counts to equivalent thermistor temperatures

module: ad_aquis()

description: The following equation is used to convert the AI channel counts to the temperatures as sensed by the thermistors.

$$\text{Temp. (Celsius)} = \frac{3825}{\ln\left(\frac{\text{channel count}}{34.86\ I}\right)} - 273 \quad \text{, where I = constant\_current in Amps}$$

B.8 Open a data file called "testrslt.dat" and write to it the relevant time(s), and the flow and thermistor temperature values

module: openfile()

description: A file is opened on the hard drive of the PC running the program. The values for the average flow rate, the opening and closing times of the valves and the cylinder temperatures are written to it, together with a timestamp.

B.9 Determine the present time from the PC system clock

module: time()

description: The value of real time in hour:min:sec can be determined from the PC system clock by using the appropriate 'C' commands.

B.10 Setup the A/D board parameters

module: flowmeter_input()

description: Prior to inputting or outputting data the software needs to set up a number of parameters which tell the A/D board what task to perform and how to go about it. For digital input the array elements 17 to 35 of param[] need to be specified.

B.11 Scan the DI channel once every 10 seconds to obtain the pulse counts

module: flowmeter_input() and timeloop()

description: During the flow stage of the test cycle a flow reading once every 10 seconds was used. The module timeloop() loops on itself while checking the PC system clock for the elapse of 10.0 seconds; it then allows flowmeter_input() to scan the AI channels of the A/D board.

B.12 Convert the channel counts to an equivalent total flow value in litres

module: flowmeter_input()

description: The conversion takes the form of a 'C' language statement , %u, which is applied to digin[0] in order to obtain a decimal integer value.

B.13 If the total volume drained out of the cylinder equals the desired outflow than shut the valves

module: flowmeter_input()

description: Test to see if the total amount of hot water that has flowed out of the cylinder is equivalent to or greater than the setup parameter value assigned to 'outflow_desired'. If this is the case than call on the module for closing the valves.

B.14 Calculate the average flow rate

module: flowmeter_input()

description: The opening time of the valves is converted to an equivalent number of seconds, the resultant value is deducted from the closing time, also in seconds, and the result is divided by 60 in order to arrive at the length of time in minutes that the valves have been open. The total flow, as read by the flowmeter, is divided by this open time to finally give the average flow rate in litres/minute.

B.15 When the valves are opened or closed the time of occurrence is assigned to a variable

module: flowmeter_input()

description: The PC system clock's actual time can be accessed by a 'C' command. At the moment of opening or closing the valves the actual event time, in hours, minutes, and seconds, is saved to the respective variables of opening_time_hour, opening_time_min, opening_time_sec, closing_time_hour, closing_time_min, and closing_time_sec.

B.16 Setup the A/D board parameters

module: close_valves()

description: Prior to inputting or outputting data the software needs to set up a number of parameters which tell the A/D board what task to perform and how to go about it. For digital output the array elements 33 to 38 of param[] need to be specified.

B.17 Select the correct DO

module: close_valves()

description: The value of digout[0] is set to 0 which results in a DO value of 0000000 being output on the 7 respective channels of the DO section.

B.18 Close the desired flow valves

module: close_valves()

description: Each flow valve is connected to a relay, which is operated directly by the DO section of the A/D board. The value of digout[0] = 0 will open each of the relays connected to the 7 output channels, which in turns cuts the 240V power to the valves, closing all of the ones that were open.

Data dictionary: data items related to the HW cylinder data acquisition program

average_flowrate = floating point value; *the average flowrate of the hot water from the cylinder in litres per minute.*
closing_time_hour = t.ti_hour'
closing_time_min = t.ti_min
closing_time_sec = t.ti_sec
constant_current = floating point value; *the value of the current flowing through the thermistors in mA, normally set at 0.19mA.*
data[100]; *the array used for storing the A/D conversion data.*
{digit} = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9.
digin[0]; *the first element of the array digin[] stores the count from the D/I, which in this case is the number of pulses from the flowmeter (note that a 'C' language conversion specification, %u, has to be applied to digin[0] in order to obtain a useful decimal integer).*
digin[10]; *the array used for storing the D/I data.*
digout[0] = {digit}$_1^2$; *the first element of the array digout[] controls the D/O relays, for instance a value of 0 switches all elements off or closes all valves by outputting 00000000 on D/O channels 0 - 7 or 8 -15.. The values range from 0 to 31. Their actions are as follows:*

0 - deactivates all relays, closes valves, switches off heating elements.
1 - switches on the upper heating element.
2 - switches on the lower heating element.
3 - opens the outlet valve and valve A.
5 - opens the outlet valve and valve B.
7 - opens the outlet valve and valve A + B.
9 - opens the outlet valve and valve C.
11 - opens the outlet valve and valve A + C.
13 - opens the outlet valve and valve B + C.
15 - opens the outlet valve and valve A + B + C.
17 - opens the outlet valve and valve D.
19 - opens the outlet valve and valve A + D.
21 - opens the outlet valve and valve B + D.
23 - opens the outlet valve and valve A + B + D.
25 - opens the outlet valve and valve C + D.
27 - opens the outlet valve and valve A + C + D.
29 - opens the outlet valve and valve B + C + D.
31 - opens the outlet valve and valve A + B + C + D.

digout[10]; *the array used for storing the D/O data.*
element_on = 0 | 1; *a flag which indicates whether a heating element has been switched on (=1) or off (=0).*
flowrate_desired ={digit}$_1^2$; *the rate of water to flow out of the cylinder, ranging from 1 to 15 litres per minute.*
HW_height = floating point value; *the height in metres of the column of water to be heated, value not to exceed 1 metre.*
looped_5_times = {digit};
opening_time_hour = t.ti_hour

opening_time_min = t.ti_min
opening_time_sec = t.ti_sec
outflow_desired ={digit}$_1^3$; *the volume of water to flow out of the cylinder, ranging from 1 to 180 litres.*
param[0] = 0 | 1; *defines the A/D board number, up to two boards allowed.*
param[1] = 0x200; *the A/D board I/O address.*
param[2] = 1 | 2 | 3; *specifies the DMA channel for the buffer.*
param[4] = 1 | 2 | 3; *specifies the IRQ channel.*
param[9] = 1; *the number of times each A/D channel is to be read.*
param[10] = FP_OFF(pointer to array); *the offset of the Analogue Input data array address.*
param[11] = FP_SEG(pointer to array); *the segment of the Analogue Input data array address.*
param[12] = 0 | 1; *if only one Analogue Input buffer is used put 0, for the use of both buffers put 1.*
param[14] ={digit}$_1^4$; *the number of A/D conversions that need to be made, ranging from 1 to 16.*
param[15] = {digit}$_1^3$; *the A/D input channel number where the conversions need to start, ranging from 0 to 15.*
param[16] = {digit}$_1^2$; *the A/D input channel number where the conversions need to stop, ranging from 0 to 15.*
param[17] = digit; *the gain code for the A/D channels, ranging from 0 to 7 (e.g. 7; 0 - 1.25V).*
param[27] = FP_OFF(pointer to array); *the offset of the Digital Input data array address.*
param[28] = FP_SEG(pointer to array); *the segment of the Digital Input data array address.*
param[29] = 0 | 1; *if only one Digital Input buffer is used put 0, for the use of both buffers put 1.*
param[31] = {digit}$_1^5$; *the number of Digital Input readings that need to be made, ranging from 1 to 65535.*
param[32] = 0 | 1; *if only one Digital Input buffer is used put 0, for the use of both buffers put 1.*
param[33] = FP_OFF(pointer to array); *the offset of the Digital Output data array address.*
param[34] = FP_SEG(pointer to array); *the segment of the Digital Output data array address.*
param[35] = 0 | 1; *if only one Digital Output buffer is used put 0, for the use of both buffers put 1.*
param[37] = {digit}$_1^5$; *the number of Digital Outputs that need to be made, ranging from 1 to 65535.*
param[38] = 0 | 1; *Digital Output Block number, D/O 0 - 7 = 0, D/O 8 - 15 = 1.*
quantity_desired ={digit}$_2^3$; *the volume of water to be heated, ranging from 20 to 180 litres.*
second_divider = 10.0 | 59.9; *this is used in the timeloop to delay by 10 or 60 seconds.*
t.ti_hour = {digit}$_2^2$; *a C language structure 't' for time in hours, ranging from 00 to 23.*
t.ti_min = {digit}$_2^2$; *a C language structure 't' for time in minutes, ranging from 00 to 59.*
t.ti_sec = {digit}$_2^2$; *a C language structure 't' for time in seconds, ranging from 00 to 59.*
temp_desired ={digit}$_2^2$; *the temperature that the water should be heated to, ranging from 10 to 80 degrees Celsius.*

Th_n ={digit}$_1^2$; *the number of the thermistor located closest to the bottom boundary of the column of water to be heated, ranging from 0 to 18.*

Th_upper_element = 7; *identifies thermistor no.7 on the strip as being nearest to the upper heating element.*

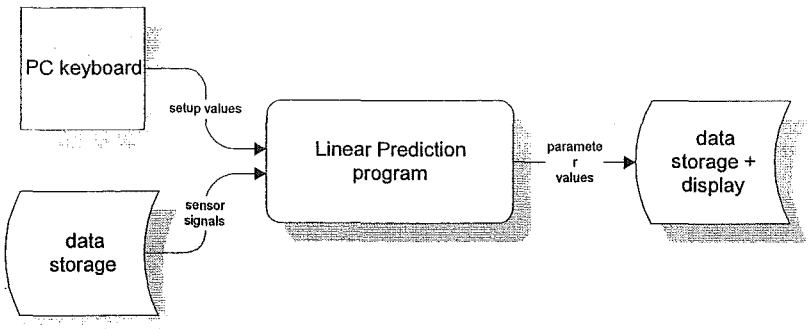
total_closing_time_sec = floating point value; *valve(s) opening time converted to seconds.*

total_flowtime_min = floating point value; *the total time that hot water has been flowing out of the cylinder in minutes.*

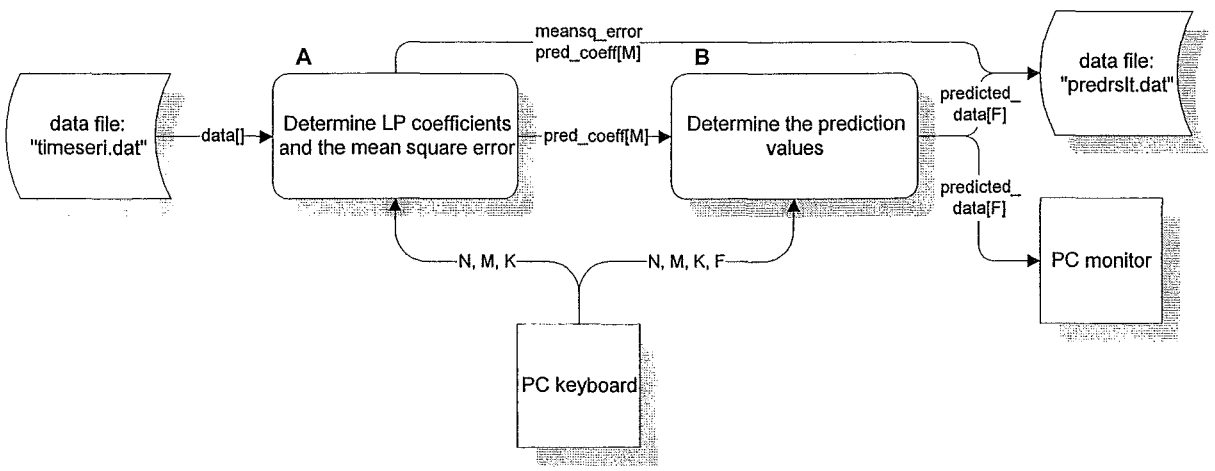
total_opening_time_sec = floating point value; *valve(s) closing time converted to seconds.*

Hot Water Cylinder Linear Prediction Program

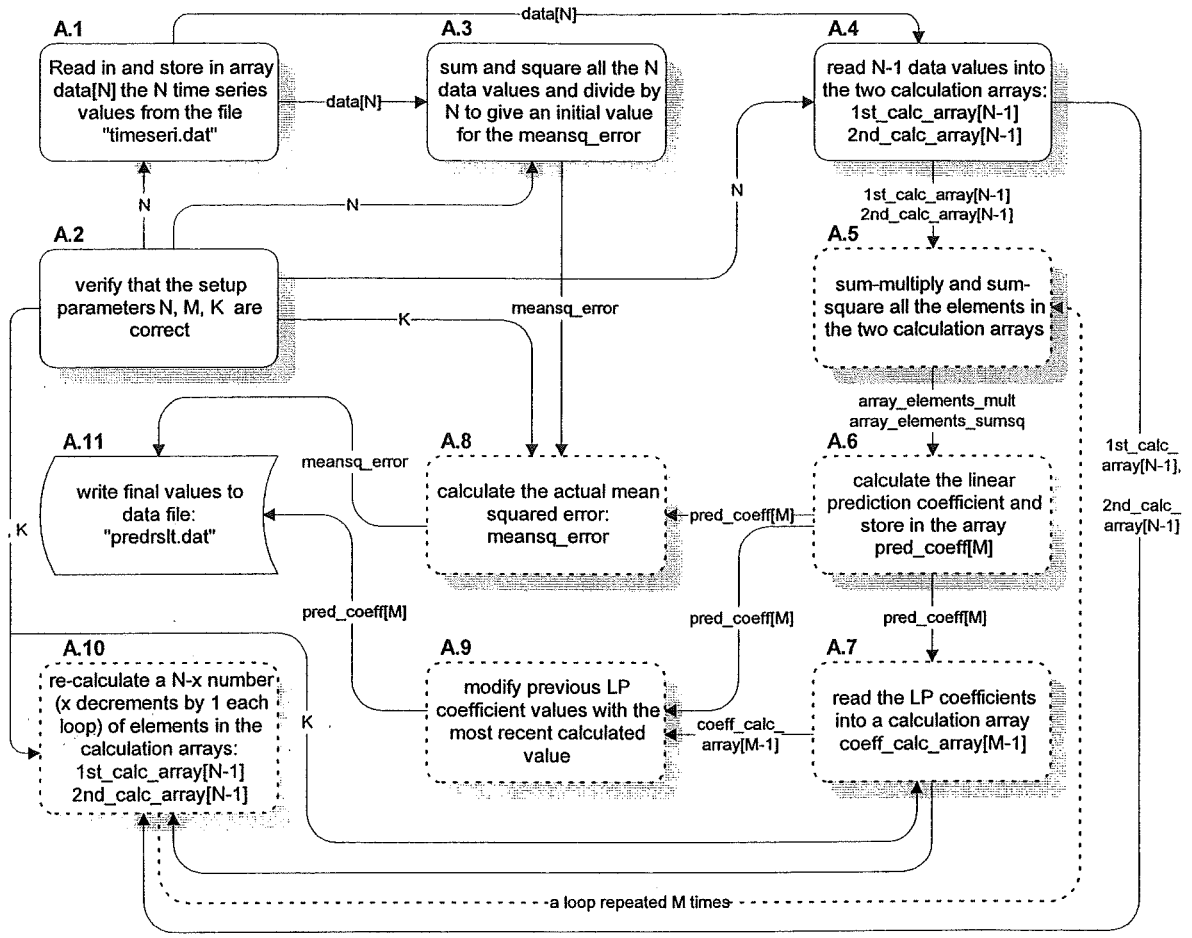
High Level Data Flow diagram



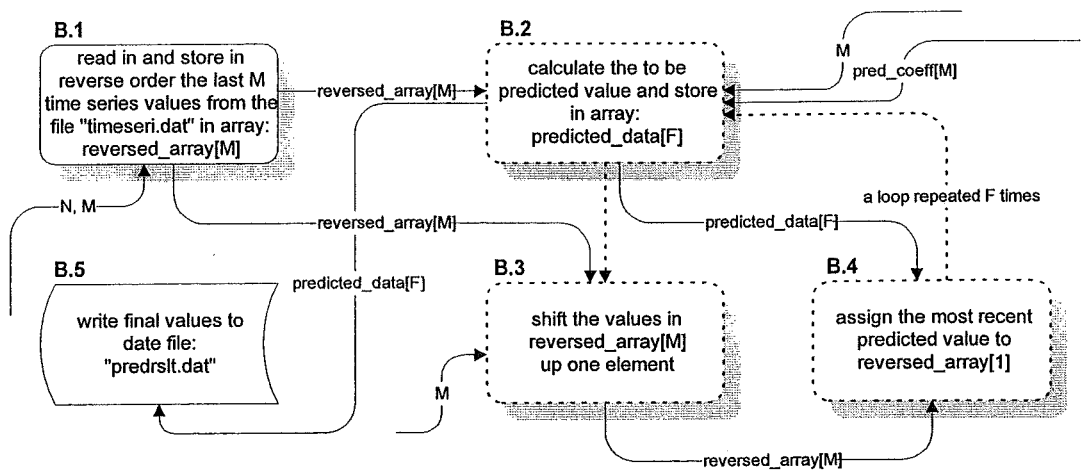
Level 1 Data Flow diagram

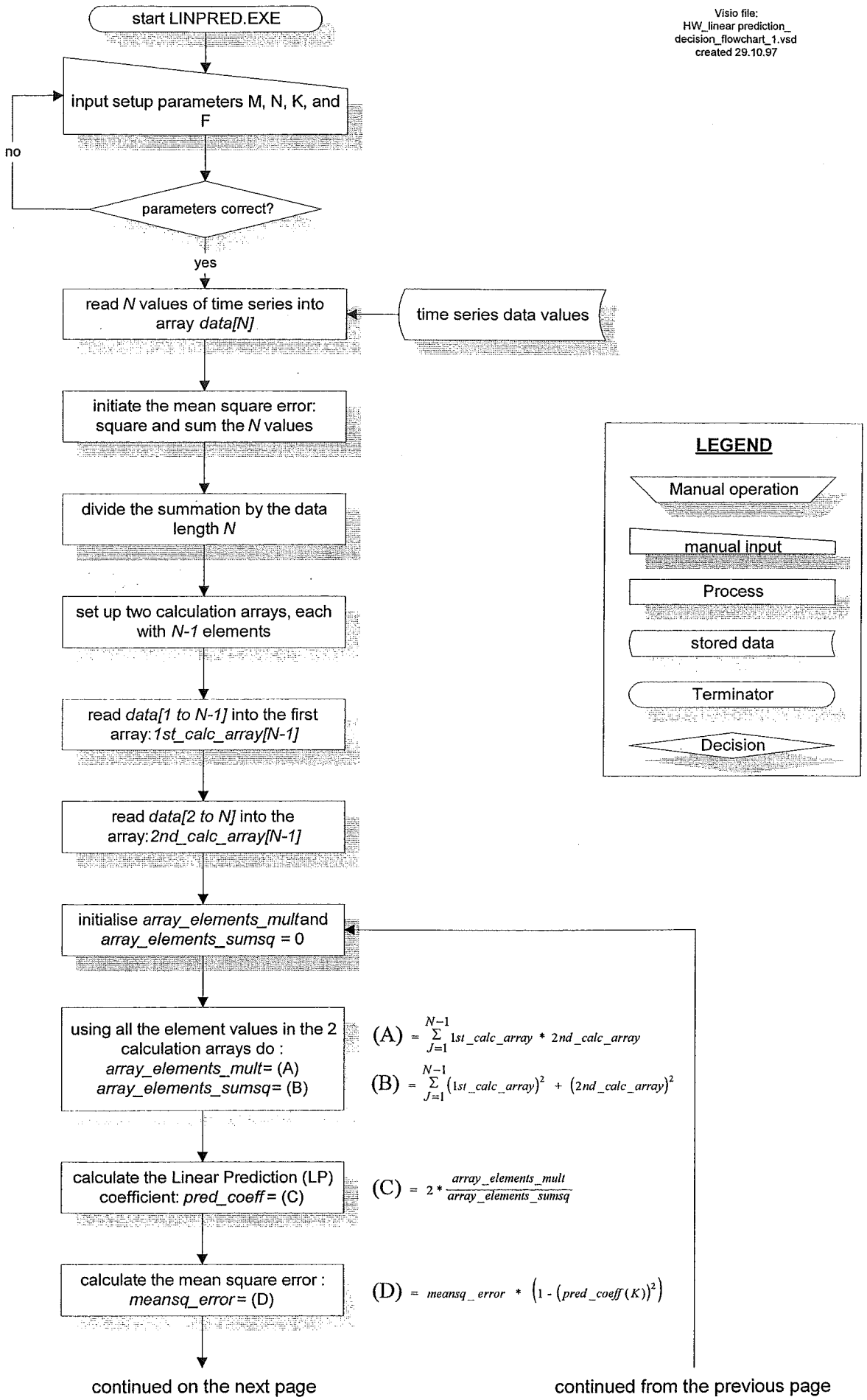


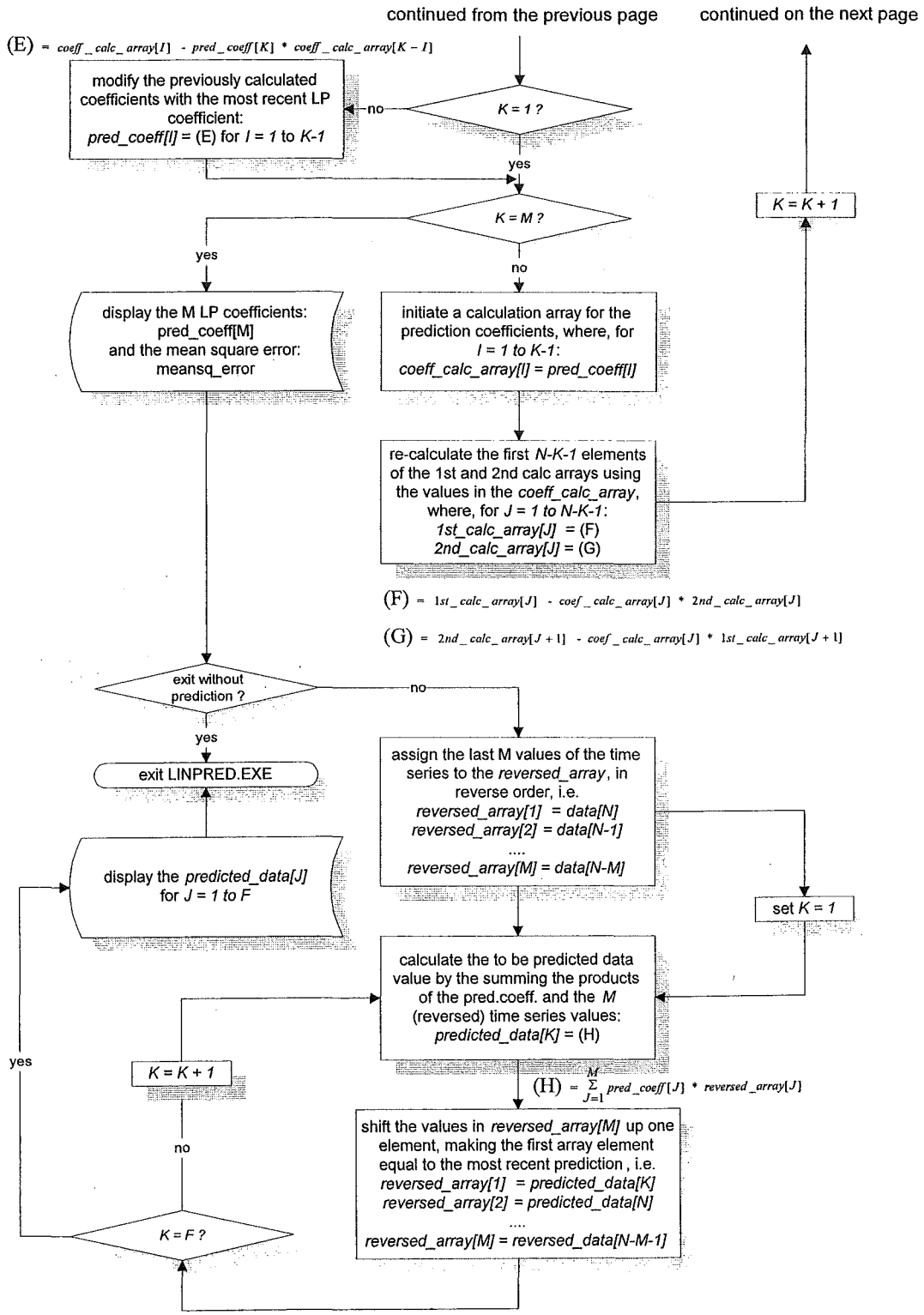
Level 2 Data Flow diagram for 'A'



Level 2 Data Flow diagram for 'B'




start LINPRED.EXE
Visio file:
HW_linear prediction_
decision_flowchart_1.vsd
created 29.10.97
input setup parameters M, N, K, and F
no
parameters correct?
yes
read N values of time series into array data[N]
time series data values
initiate the mean square error: square and sum the N values
divide the summation by the data length N
set up two calculation arrays, each with N-1 elements
read data[1 to N-1] into the first array: 1st_calc_array[N-1]
read data[2 to N] into the array: 2nd_calc_array[N-1]
initialise array_elements_mult and array_elements_sumsq = 0
using all the element values in the 2 calculation arrays do :
array_elements_mult= (A)
array_elements_sumsq= (B)
(A) = $\sum_{J=1}^{N-1} 1st\_calc\_array * 2nd\_calc\_array$
(B) = $\sum_{J=1}^{N-1} (1st\_calc\_array)^2 + (2nd\_calc\_array)^2$
calculate the Linear Prediction (LP) coefficient: pred_coeff = (C)
(C) = $2 * \frac{array\_elements\_mult}{array\_elements\_sumsq}$
calculate the mean square error : meansq_error = (D)
(D) = $meansq\_error * \left(1 - (pred\_coeff(K))^2\right)$
continued on the next page
continued from the previous page
LEGEND
Manual operation
manual input
Process
stored data
Terminator
Decision



A flow diagram that shows the activities and decisions undertaken by the software for linear prediction.

Activity specification: description of the basic activities for the Linear Prediction algorith

A.1 Read in and store in array *data[N]* the *N* time series values from the file "timeseri.dat"

module: linpred_coeff()

description: The .dat file contains the values that make up the HW usage time series, i.e. the daily amount of water used expressed in litres. These values are stored in an array in order to make them accessible for program manipulation.

A.2 Verify that the setup parameters *N, M, K* are correct

module: linpred_coeff()

description: The user is asked to check and confirm the setup parameters entered via the keyboard. If not correct, alterations to the values can be made.

A.3 Sum and square all the *N* data values and divide by *N* to give an initial value for the meansq_error

module: linpred_coeff()

description:

$$sumsq\_error = \frac{\sum_{J=1}^{N}(data[J])^2}{N}$$

A.4 Read N-1 data values into the two calculation arrays: *1st_calc_array[N-1]* and *2nd_calc_array[N-1]*

module: linpred_coeff()

description: Two 'calculation' arrays are needed which can act as 'notepads', allowing values to be stored and worked upon. On initialisation the first array stores the data series values from the *first* to the *N*-1 (second-last) figure, the second array contains the values from the *second* figure to the *N*th (last) figure in the time series.

A.5 Sum-multiply and sum-square all the elements in the two calculation arrays

module: linpred_coeff()

description: Using all the element values in the 2 calculation arrays the following two variables are arrived at :

$$array\_elements\_mult = \sum_{J=1}^{N-1} 1st\_calc\_array \; * \; 2nd\_calc\_array$$

$$array\_elements\_sumsq = \sum_{J=1}^{N-1} (1st\_calc\_array)^2 \; + \; (2nd\_calc\_array)^2$$

A.6 Calculate the linear prediction coefficient and store in the array *pred_coeff[M]*

module: linpred_coeff()

description: The LP coefficient is calculated as follows:

$$pred\_coeff\ [M] = 2 * \frac{array\_elements\_mult}{array\_elements\_sumsq}$$

where the first calculated value is stored in the first element of the array, and further calculated values get stored in subsequent elements; which occurs for every one of the M loops the program makes.

A.7 Read the LP coefficients into a calculation array *coeff_calc_array[M-1]*

module: linpred_coeff()

description: initiate a calculation array for the prediction coefficients, where, for $I = 1$ to K-1:

coeff_calc_array[I] = pred_coeff[I]

A.8 Calculate the actual mean squared error: *meansq_error*

module: linpred_coeff()

description: $meansq\_error = meansq\_error * \left(1 - \left(pred\_coeff(K)\right)^2\right)$

where the initial value for the error comes from activity A.3.

A.9 Modify previous LP coefficient values with the most recent calculated value

module: linpred_coeff()

description: modify the previously calculated coefficients with the most recent LP coefficient:

for $I = 1$ to K-1, *pred_coeff[I]* =

$coeff\_calc\_array[I] - pred\_coeff[K] * coeff\_calc\_array[K - I]$

.

A.10 Re-calculate a N-x number (x decrements by 1 each loop) of elements in the calculation arrays: *1st_calc_array[N-1]* and *2nd_calc_array[N-1]*

module: linpred_coeff()

description: re-calculate the first N-K-1 elements of the 1st and 2nd calc arrays using the values in the coeff_calc_array, where, for $J = 1$ to N-K-1:

1st_calc_array[J] =

$1st\_calc\_array[J] - coef\_calc\_array[J] * 2nd\_calc\_array[J]$

2nd_calc_array[J] =

$2nd\_calc\_array[J+1] - coef\_calc\_array[J] * 1st\_calc\_array[J+1]$.

A.11 Write final values to data file: "*predrslt.dat*"

module: linpred_coeff()

description: When all M values for the LP coefficients have been calculated and the final mean square error is known, than these figures are written to a data file (and displayed on the monitor).

B.1 Read in and store in reverse order the last *M* time series values from the file "*timeseri.dat*" in array: *reversed_array[M]*

module: prediction()

description: assign the last *M* values (*M* being equivalent to the number of LP coefficients) of the time series to the *reversed_array[M]*, in reverse order, i.e.

reversed_array[1] = data[N]
reversed_array[2] = data[N-1]
....
reversed_array[M] = data[N-M].

B.2 Calculate the to be predicted value and store in array: *predicted_data[F]*

module: prediction()

description: calculate the to be predicted data value by the summing the products of the prediction coefficients and the *M* (reversed) time series values:

$$predicted\_data[K] = \sum_{J=1}^{M} pred\_coeff[J] \ * \ reversed\_array[J]$$

where the value of *K* starts at 1 and is incremented by one at a time until *F* loops are completed.

B.3 Shift the values in *reversed_array[M]* up one element

module: prediction()

description: The *reversed_array[M]* acts as a 'notepad'; as such the array values now need to shift up one element, i.e. in the first of *F* loops this would mean that:

reversed_array[2] = predicted_data[N]
reversed_array[3] = predicted_data[N-1]
....
reversed_array[M] = reversed_data[N-M-1]

B.4 Assign the most recent predicted value to *reversed_array[1]*

module: prediction()

description: Having shifted the values up one element in B.3, need to (re)define the first array element equal to the most **recent** prediction, i.e.

reversed_array[1] = predicted_data[K].

B.5 write final values to date file: "*predrslt.dat*"

module: prediction()

description: When all *F* prediction values have been, than these figures are written to a data file (and displayed on the monitor).

Data dictionary: data items related to the HW Linear Prediction Program.

data[N]; *the array used for storing the time series data.*

array_elements_mult = floating point value; *stores the value obtained by summing the multiplied 'calculation array' elements.*

array_elements_sumsq = floating point value; *stores the value obtained by summing the sum- squared 'calculation array' elements.*

pred_coeff [M]; *the array used for storing the M individually calculated linear prediction coefficients.*

{digit} = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9.

coeff_calc_array [M-1]; *the array repeatedly used for storing previously calculated linear prediction coefficients.*

reversed_array[M]; *the array used to store the last M values of the time series; these values get stored in a reverse order to what they were in the time series.*

predicted_data[F]; *the array that holds the calculated values of the prediction.*

F = {digit}$_1^2$; *the number of data points that are to be predicted.*

N = {digit}$_1^4$; *the number of data values in the time series under consideration.*

1st_calc_array [N-1]; *the array repeatedly used for storing calculations initially based on the **first** N-1 values of the time series data.*

2nd_calc_array [N-1]; *the array repeatedly used for storing calculations initially based on the **last** N-1 values of the time series data.*

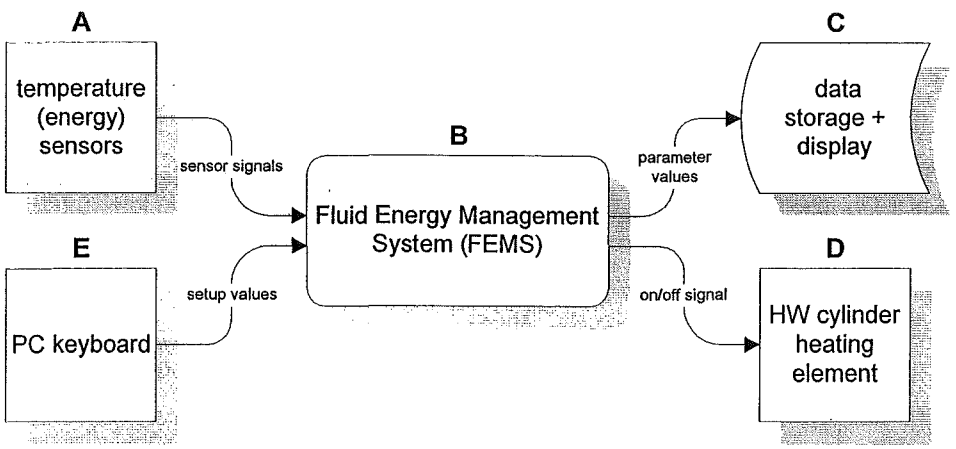
meansq_error = floating point value; *this variable holds the mean squared error.*

M = {digit}$_1^3$; *the number of linear prediction coefficients being determined.*

I, J, K = {digit}$_1^3$; *counter values which are reset/incremented where necessary.*

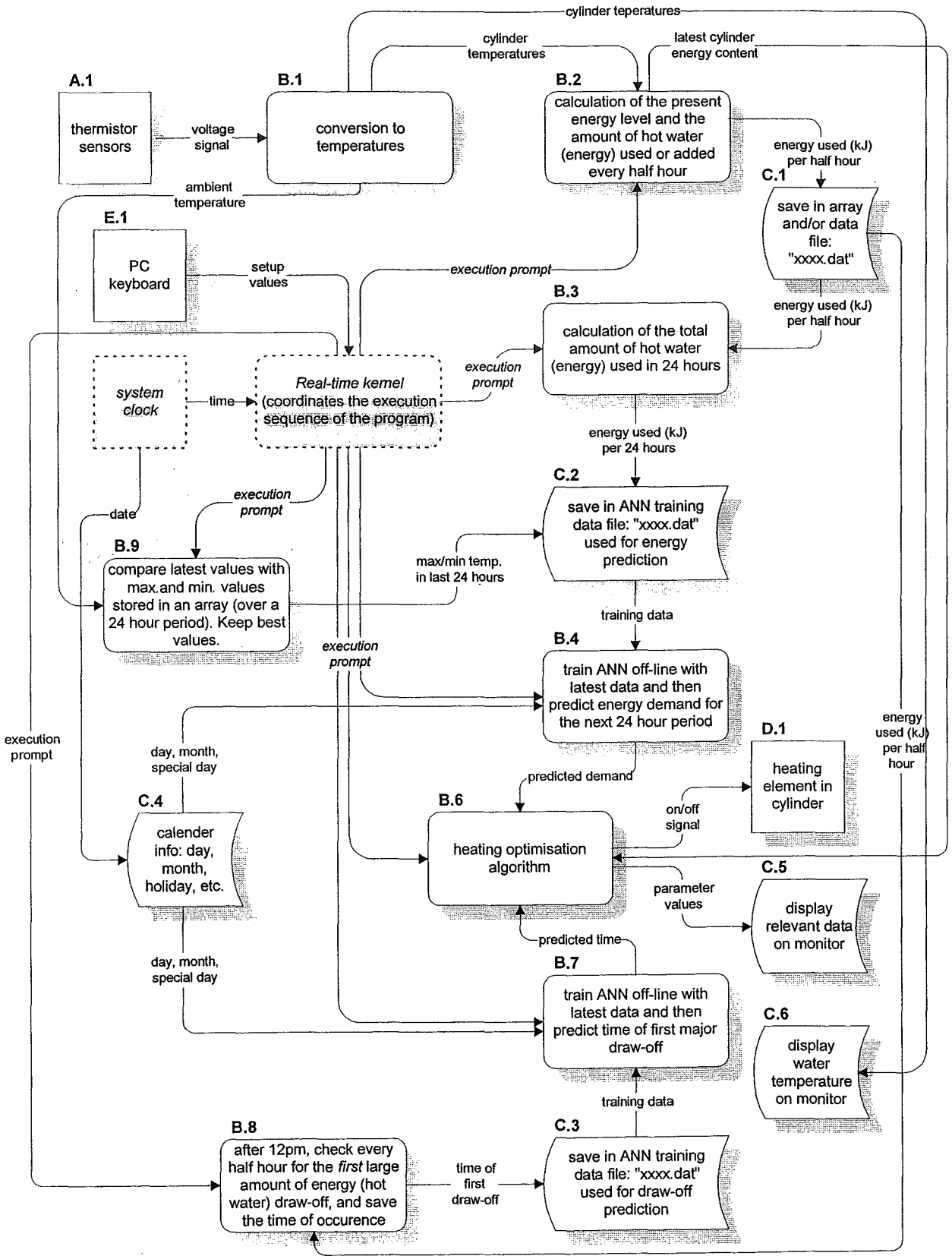
Hot Water Cylinder Fluid Energy Management Program

High Level Data Flow diagram



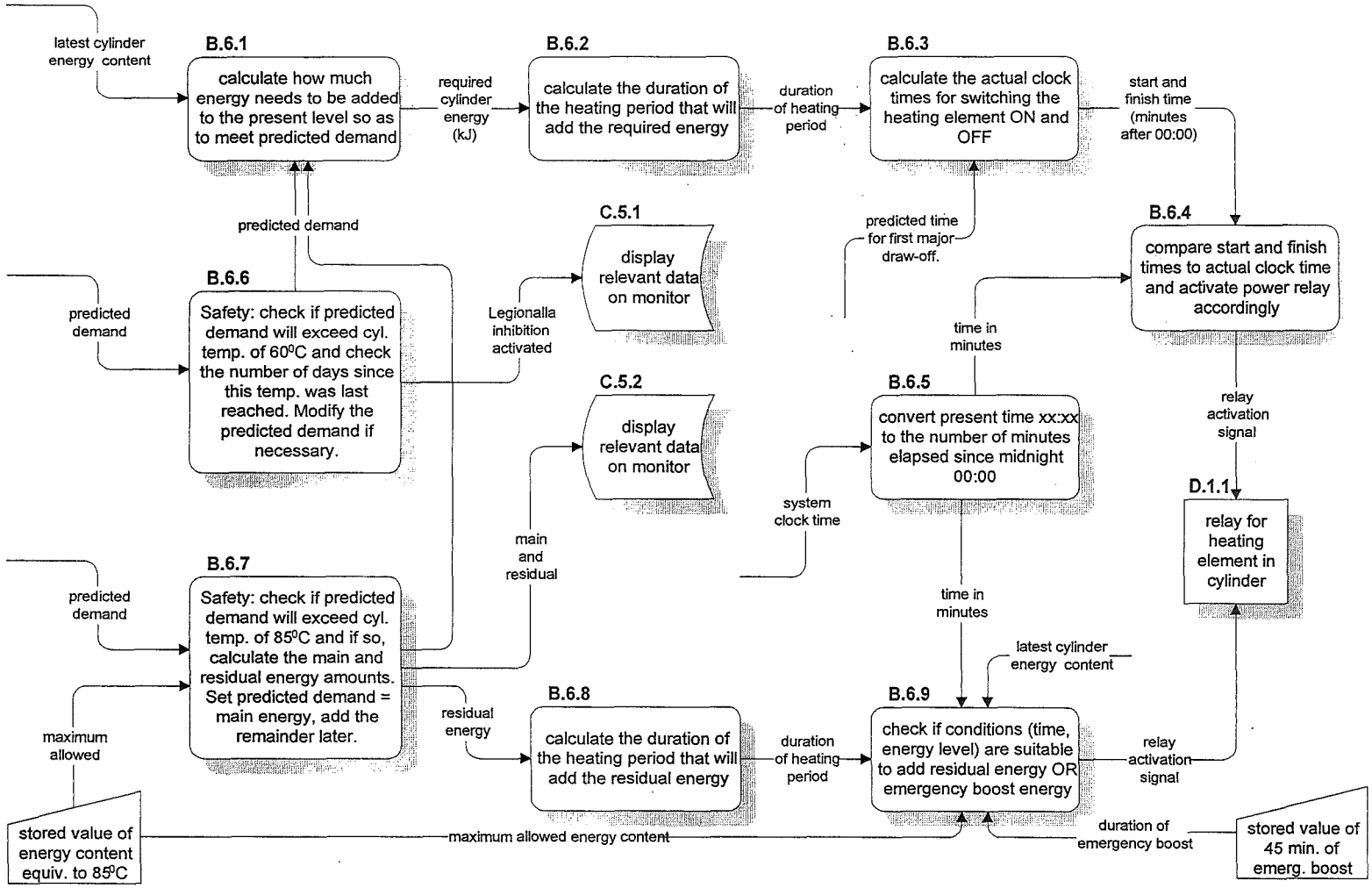
Level 1 Data Flow diagram - FEMS

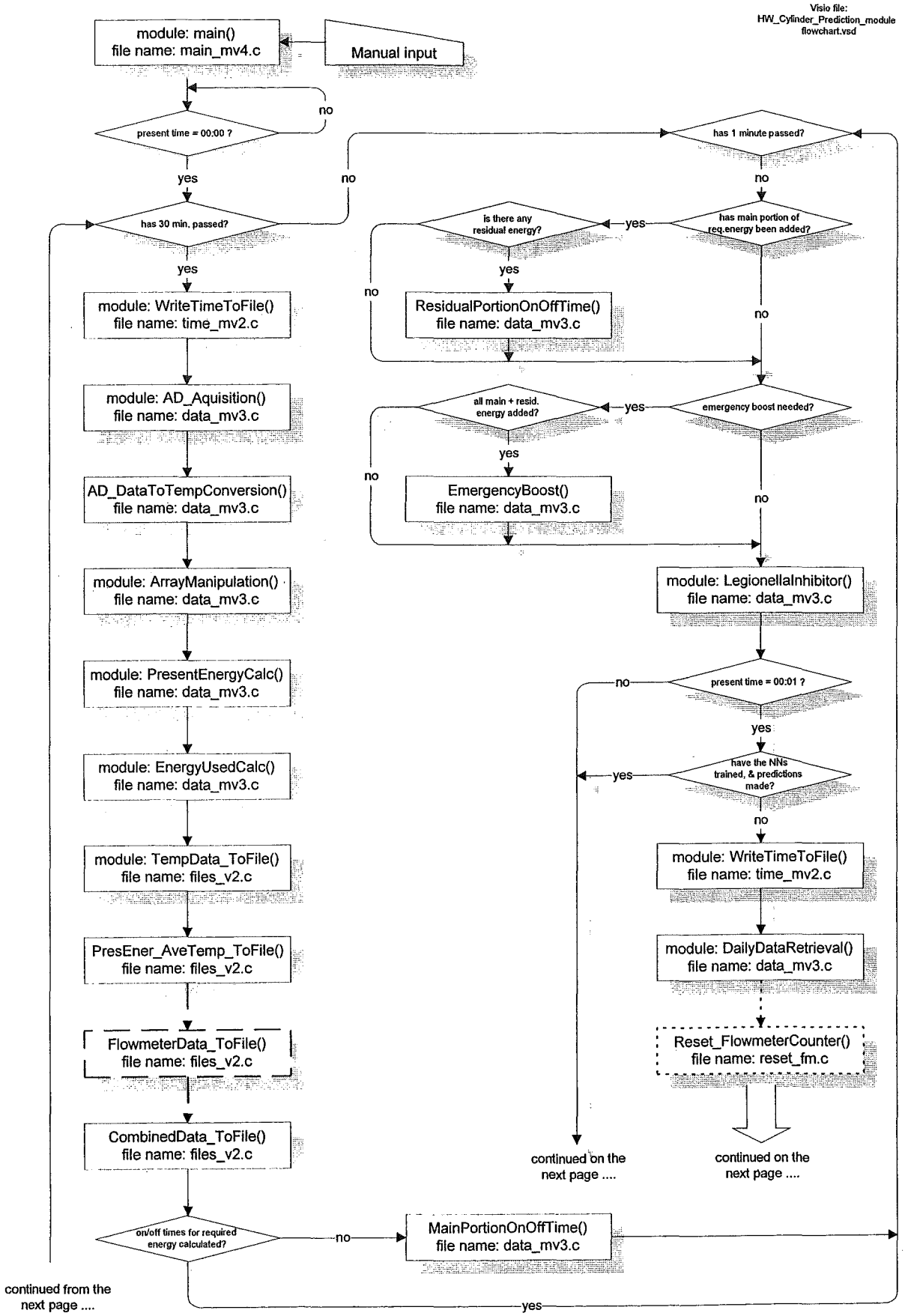
file name: HW_ANN Prediction_dataflow_all levels

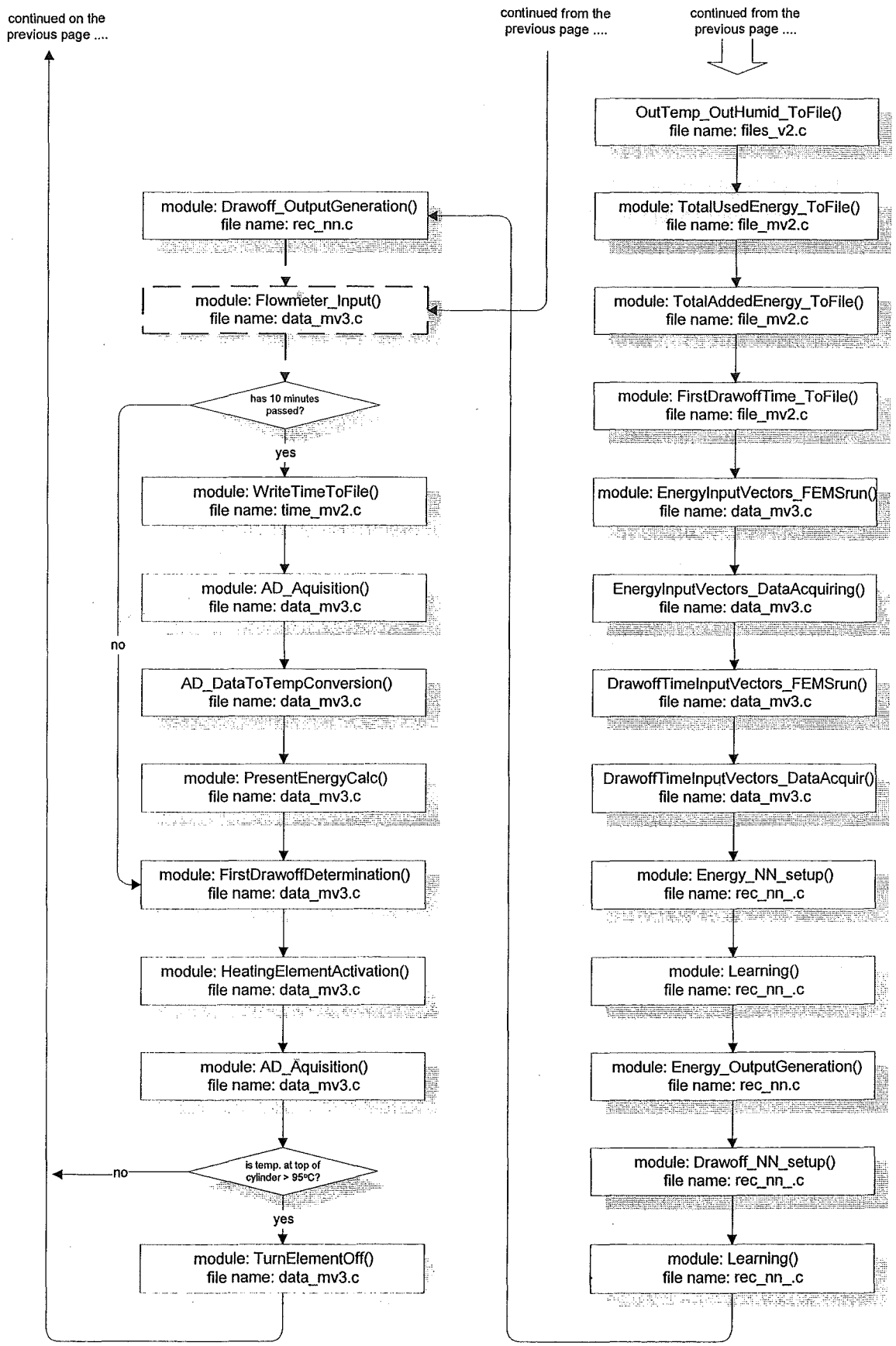


Level 2 Data Flow diagram - FEMS of the 'heating optimisation algorithm' activity B.6

file name: HW_ANN Prediction_dataflow_all levels




Visio file:
HW_Cylinder_Prediction_module
flowchart.vsd
module: main()
file name: main_mv4.c
Manual input
present time = 00:00 ?
no
yes
has 30 min. passed?
no
yes
module: WriteTimeToFile()
file name: time_mv2.c
module: AD_Aquisition()
file name: data_mv3.c
AD_DataToTempConversion()
file name: data_mv3.c
module: ArrayManipulation()
file name: data_mv3.c
module: PresentEnergyCalc()
file name: data_mv3.c
module: EnergyUsedCalc()
file name: data_mv3.c
module: TempData_ToFile()
file name: files_v2.c
PresEner_AveTemp_ToFile()
file name: files_v2.c
FlowmeterData_ToFile()
file name: files_v2.c
CombinedData_ToFile()
file name: files_v2.c
on/off times for required energy calculated?
no
yes
MainPortionOnOffTime()
file name: data_mv3.c
continued from the next page ....
has 1 minute passed?
no
yes
has main portion of req.energy been added?
yes
no
is there any residual energy?
yes
no
ResidualPortionOnOffTime()
file name: data_mv3.c
emergency boost needed?
yes
no
all main + resid. energy added?
yes
no
EmergencyBoost()
file name: data_mv3.c
module: LegionellaInhibitor()
file name: data_mv3.c
present time = 00:01 ?
no
yes
have the NNs trained, & predictions made?
yes
no
module: WriteTimeToFile()
file name: time_mv2.c
module: DailyDataRetrieval()
file name: data_mv3.c
Reset_FlowmeterCounter()
file name: reset_fm.c
continued on the next page ....
continued on the next page ....



A flow diagram for the 'C' modules as used in the software for testing water behaviour in a hot water cylinder.

*DeltaWeightsPointer [NMXHLR+1] ;
*ErrorPointer [NMXHLR+2] ;
*OutputPointer [NMXHLR+2] ;
*WeightsPointer [NMXHLR+1] ;
{digit} = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9.
alpha = {digit}$_1^4$; *momentum rate.*
AverageTemp = {digit}$_1^2$; *average temperature of the cylinder as measured by the thermistors.*
constant_current = floating point value; *the value of the current flowing through the thermistors in mA, normally set at 0.19mA.*
CONTNE = 0; *continue calculation.*
data[100] = {digit}$_1^5$; *the array used for storing the A/D conversion data.*
data1[20] = {digit}$_1^5$; *array that holds the A/D count from first board.*
data2[20] = {digit}$_1^5$; *array that holds the A/D count from second board.*
degrees_celsius = {digit}$_1^2$; *ambient temperature is measured in degrees Celsius and can range from -20 – 40.*
digin[0] = {digit}$_1^2$; *the first element of the array digin[] stores the count from the D/I, which in this case is the number of pulses from the flowmeter (note that a 'C' language conversion specification, %u, has to be applied to digin[0] in order to obtain a useful decimal integer).*
digin[10] = {digit}$_1^2$; *the array used for storing the D/I data.*
digin[2] = {digit}$_1^2$; *digital input data array as used for the flowmeter.*
digout[0] = {digit}$_1^2$; *the first element of the array digout[] controls the D/O relays, for instance a value of 0 switches all elements off.*
digout[10] = {digit}$_1^2$; *the array used for storing the D/O data.*
DRAWOFF_ANN_TRAINING_FLAG = 0 | 1; *a flag allowing a sub-routine (module) to be used.*
drawoff_input_vector[] = "drawpred"; *filename for the predicted drawoff data.*
DRAWOFF_PREDICTION_FLAG = 0 | 1; *a flag allowing a sub-routine (module) to be used.*
drawoff_task_name[] = "hisdrw"; *filename for the historic drawoff data.*
DrawTimePredInputVector[29] = {digit}$_1^5$; *as above, but then values will be used for predicting the first draw off time.*
EMERGENCY_BOOST_USED = 0 | 1; *a flag allowing a sub-routine (module) to be used.*
ENERGY_ANN_TRAINING_FLAG = 0 | 1; *a flag allowing a sub-routine (module) to be used.*
energy_array = {digit}$_1^2$; *this array contains all the total energy consumption values up to a maximum of the last 84 days.*
ENERGY_PREDICTION_FLAG = 0 | 1; *a flag allowing a sub-routine (module) to be used.*

EnergyAdded[48] = {digit}$_1^5$; *a 48-element array which solely holds any added energy values from the last 24hrs.*

EnergyContent[2] = {digit}$_1^5$; *a two-element array which holds the last two energy contents. The values could be negative if the temperature of the cylinder drops below 40 degr.C. and PresentEnergyContent is a neg. value.*

EnergyPredInputVector[29] = {digit}$_1^5$; *this array contains NORMALIZED values for an energy prediction input vector.*

EnergyScalingFactor = {digit}$_1^5$; *any calculated or measured energy value is divided by this value for scaling.*

EnergyUsed[48] = {digit}$_1^5$; *a 48-element array which holds 24hrs worth of used energy values. The values could be negative if energy is added to the cylinder. It is designated 'long' because the value can exceed +/- 32,767.*

ep[MAXNOINPUTVECTORS] ;

eta = {digit}$_1^4$; *learning rate.*

FEXIT = 1; *exit in failure.*

FileInputData [MAXNOINPUTVECTORS] [NMXIATTR] ,

FileTargetData [MAXNOINPUTVECTORS] [NMXOATTR] ;

finish_time_element_on = minutes_after_midnight; *the time after midnight when the element is switched off.*

FirstDrawoffTime = {digit}$_1^4$; *a variable that contains the time that the first draw off took place.*

half_hour_previous_cylinder_energy_content = kilo_Joule_amount; *the energy present in the cylinder as read half an hour previously.*

half_hourly_ambient_temperature = degrees_celsius; *the latest half hourly temperature reading.*

half_hourly_consumed_energy = kilo_Joule_amount; *the reduction in energy as measured each half hour.*

holiday_boolean = 0 | 1; *single digit input for holiday.*

IterationCounter = {digit}$_1^4$; *keeps track of the number of epochs during training.*

kilo_Joule_amount = {digit}$_1^5$; *an amount of energy, range from 0 – 54,000.*

Latest_Flowmeter_Reading = digin[0]; *Contains the value from digin[0].*

LEGIONELLA_INHIBITION_USED = 0 | 1; *a flag allowing a sub-routine (module) to be used.*

length_time_element_on = minutes; *the length of time needed for the element to add the required energy.*

MAIN_PORTION_FLAG = 0 | 1; *a flag allowing a sub-routine (module) to be used.*

MaxAllowedEnergyContent = 32980; *the maximum energy (kJ) a consumer is safe to store and extract from the HW cylinder. In this case 85 degr.C. and 180 litres.*

maxe = {digit}$_1^4$; *maximum allowed neural network error.*

maxep = {digit}$_1^4$; *maximum allowed pattern error.*

maximum_24hour_temperature = degrees_celsius; *the maximum ambient temperature over the last 24 hour period.*

MAXNOINPUTVECTORS = {digit}$_1^2$; *maximum number of input vectors.*

MaxOutsideHumid = {digit}$_1^2$; *the value of the highest outside humidity reached in the last 24 hours.*

MaxOutsideTemp = {digit}$_1^2$; *the value of the highest outside temperature reached in the last 24 hours.*

minimum_24hour_temperature = degrees_celsius; *the minimum ambient temperature over the last 24 hour period.*

MinOutsideHumid = {digit}$_1^2$; *the value of the lowest outside humidity reached in the last 24 hours.*

MinOutsideTemp = {digit}$_1^2$; *the value of the lowest outside temperature reached in the last 24 hours.*

minutes = {digit}$_1^3$; *a length of time designated in minutes, range 1 – 180.*

minutes_after_midnight = {digit}$_1^4$; *the time as measured in the total no. of minutes after midnight, range from 0 - 1439.*

MONITOR_DRAWOFF_FLAG = 0 | 1; *a flag allowing a sub-routine (module) to be used.*

month_boolean = 0 | 1; *three digit input for the month.*

NeuralNetOutput [MAXNOINPUTVECTORS] [NMXOATTR] ;

NeuronsPerLayer [NMXHLR+2] ;

NMXHLR = {digit}$_1^2$; *maximum number of hidden layers.*

NMXIATTR = {digit}$_1^2$; *maximum number of input variables in the training vector*

NMXOATTR = {digit}$_1^2$; *maximum number of output variables in the target vector.*

NMXUNIT = {digit}$_1^2$; *maximum number of units in a layer.*

NoOfDataVariablesInEachVector = {digit}$_1^2$; *the number of data variables that make up a training vector.*

NoOfDaysAfterSunday = 1 | 2 | 3 | 4 | 5 | 6 | 7 ; *a set-up paramete,r telling FEMS which day of the week it is when the program first runs.*

NoOfHiddenLayers = {digit}; *the number of recurrent hidden layers in the NN.*

NoOfInputVectorTargetData = {digit}; *the value of the (single) target vector (goal).*

NumberCylTherm = 19; *the number of thermistors installed on the H.W. cylinder.*

OutsideHumid[48]; *array that holds the 48 halfhourly relative humidities.*

OutsideTemp[48]; *array that holds the 48 halfhourly outside temperatures.*

OverallNetworkError = {digit}$_1^4$; *sum squared error as obtained during NN training.*

param[0] = 0 | 1; *defines the A/D board number, up to two boards allowed.*

param[1] = 0x200; *the A/D board I/O address.*

param[10] = FP_OFF(pointer to array); *the offset of the Analogue Input data array address.*

param[11] = FP_SEG(pointer to array); *the segment of the Analogue Input data array address.*

param[12] = 0 | 1; *if only one Analogue Input buffer is used put 0, for the use of both buffers put 1.*

param[14] ={digit}$_1^4$; *the number of A/D conversions that need to be made, ranging from 1 to 16.*

param[15] = {digit}$_1^3$; *the A/D input channel number where the conversions need to start, ranging from 0 to 15.*

param[16] = {digit}$_1^2$; *the A/D input channel number where the conversions need to stop, ranging from 0 to 15.*

param[17] = digit; *the gain code for the A/D channels, ranging from 0 to 7 (e.g. 7; 0 - 1.25V).*

param[2] = 1 | 2 | 3; *specifies the DMA channel for the buffer.*

param[27] = FP_OFF(pointer to array); *the offset of the Digital Input data array address.*

param[28] = FP_SEG(pointer to array); *the segment of the Digital Input data array address.*
param[29] = 0 | 1; *if only one Digital Input buffer is used put 0, for the use of both buffers put 1.*
param[31] = {digit}$_1^5$; *the number of Digital Input readings that need to be made, ranging from 1 to 65535.*
param[32] = 0 | 1; *if only one Digital Input buffer is used put 0, for the use of both buffers put 1.*
param[33] = FP_OFF(pointer to array); *the offset of the Digital Output data array address.*
param[34] = FP_SEG(pointer to array); *the segment of the Digital Output data array address.*
param[35] = 0 | 1; *if only one Digital Output buffer is used put 0, for the use of both buffers put 1.*
param[37] = {digit}$_1^5$; *the number of Digital Outputs that need to be made, ranging from 1 to 65535.*
param[38] = 0 | 1; *Digital Output Block number, D/O 0 - 7 = 0, D/O 8 - 15 = 1.*
param[4] = 1 | 2 | 3; *specifies the IRQ channel.*
param[9] = 1; *the number of times each A/D channel is to be read.*
predicted_time-first-drawoff = minutes; *the forecast time after midnight when the first significant draw-off is likely to take place.*
predicted_total_energy_consumption = kilo_Joule_amount; *the forecast energy consumption as output by the neural network, valid for the next 24 hours (unit kJ).*
prediction_input_vector[] = "enerpred"; *filename for the predicted energy data.*
prediction_task_name[] = "histen"; *filename for the historic energy data.*
present_cylinder_energy_content = kilo_Joule_amount; *the energy present in the cylinder (as determined at midnight 00:00).*
PresentEnergyContent = {digit}$_1^5$; *a variable that contains the latest cylinder energy content. The values could be negative if the temperature of the cylinder drops below 40 degr.C. It is equivalent to: Volume * Product * (AverageTemp - 40).*
PresentTime = minutes_after_midnight; *the present time value in minutes after midnight.*
PreviousEnergyContent = {digit}$_1^5$; *a variable that contains the second-latest cylinder energy content. The values could be negative if the temperature of the cylinder drops below 40 degr.C.*
Product = -1.675 * AverageTemp + 4214; *a graph of (specific heat capacity x density) vs temperature gives an almost straight line with eqn y=-1.675x + 4214.*
relay_activation = {digit}; *the number send to the PCL-814 board which closes the power relay contacts for the heater element.*
relay_deactivation = {digit} *the number send to the PCL-814 board which opens the power relay contacts for the heater element.*
required_cylinder_energy = kilo_Joule_amount; *the energy required by the cylinder to bring it up to the predicted amount.*
residual_energy = kilo_Joule_amount; *the energy that might be remaining if the predicted amount is greater than the cylinder's maximum allowed energy content.*
residual_length_time_element_on = minutes; *the length of time needed for the element to add the residual energy.*
RESIDUAL_PORTION_FLAG = 0 | 1; *a flag allowing a sub-routine (module) to be used.*
RESTRT = 2; *restart.*
SEXIT = 3; *exit successfully.*
special_boolean = 0 | 1; *single digit input for special day, one without a fixed date, i.e. easter, anzac.*

start_time_element_on = minutes_after_midnight; *the time after midnight when the element is switched on.*

SumLayer = {digit}$_1^2$; *summed water layers in the cylinder.*

SumTemp = {digit}$_1^3$, *summed temperature of the individual water layers in the cylinder as associated with each thermistor sensor.*

temp[20] = {digit}$_1^2$; *array that holds the 19 + 1 thermistor temperatures.*

Thermistor_temperature = {digit}$_1^2$; *temperature is measured in degrees Celsius and can range from 10 – 90.*

TimeScalingFactor = {digit}$_1^5$; *defined as per above but then for the input and output of the N.N.*

total_consumed_energy_last 24 hours = kilo_Joule_amount; *the total energy used over the past 24 hours.*

TotalAddedEnergy = {digit}$_1^5$; *the value of the total energy added in the last 24 hours.*

TotalUsedEnergy = {digit}$_1^5$; *the value of the total energy used in the last 24 hours.*

TrainingIterations = {digit}$_1^4$; *the number of training epochs selected for the NN.*

Volume = 0.18; *the H.W. cylinder volume in cubic metres.*

WaterHeatingRate = 162.1; *the rate at which the water is heated in the 180 litre cylinder with the 3kW element: 162.1 kJ per minute.*

WaterLayer = 0.05; *the waterlayer thickness in metres associated with each thermistor.*

weekday_boolean = 0 | 1; *three digit input for the day of the week.*